Computation and Applications of the Structured Singular Value

Ph.D. Thesis
of
Martin J. M. Hayes

Computation and Applications of the Structured Singular Value

Doctor of Philosophy
in
Electronic Engineering

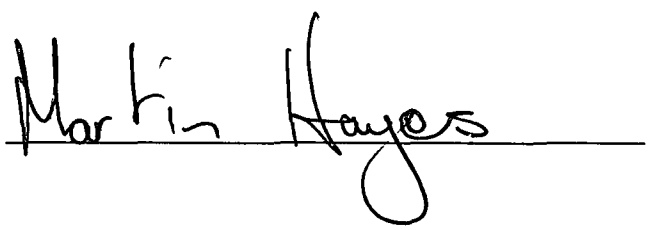
Martin J M Hayes

Dublin City University
School of Electronic Engineering

Supervised by Dr Anthony M Holohan

August 1997

I hereby certify that this material, which I now submit for assessment on the programme of study leading to the award of Doctor of Philosophy in Electronic Engineering is entirely my own work and has not been taken from the work of others save and to the extent that such work has been cited and acknowledged within the text of my work

Signed  ID No 92700519

Date 27/8/97

Acknowledgements

I would like to express my gratitude to Dr Anthony Holohan for his excellent supervision of this work His attention to detail provides a fine paradigm for the work that I would hope to supervise in the future

I would also like to thank Professor Charles McCorkell and the School of Electronic Engineering at Dublin City University Their collective wisdom in supporting doctoral research deserves much credit

Thanks are also due to Professor Cyril Burkley, Head of the Department of Electronic and Computer Engineering at the University of Limerick who allowed me the opportunity to complete this research

Finally, I realise that the temptation to make a life story out of this section should be resisted However, in addition to family and friends one person merits a special mention because she has to put up with a lot This thesis is dedicated to Anne

Contents

Abstract

This work develops a number of new algorithms for the computation of the convex estimate, μ_{co}, of the structured singular value, μ The basis for these algorithms lies in an application of the geometric form of the Hahn-Banach theorem Dual methods are shown to be more useful than their primal counterparts when determining μ_{co} In particular, a dual method based on a 1-norm optimisation strategy is shown to out perform its 2-norm based counterpart in terms of accuracy, reliability and speed It is proven that this 1-norm dual method converges to μ_{co} The algorithm has been successfully validated using a large selection of random, pseudo random and practically motivated problems Improvements to the basic algorithm that significantly reduce computation times are outlined and analysed in some detail

This work presents new applications for μ μ provides a novel way of analysing, non-conservatively, the effect of uncertainty in component values on filter performance It is shown that the problems of computing (i) maximum filter gain, (ii) minimum gain and (iii) the maximum Euclidean deviation from nominal performance on a polar plot can all be determined using μ theory The necessary formulations required to deal with these particular problems are developed In contrast to a grid search or probabilistic approach, using μ provides frequency response information that fully addresses the cross coupling effects of component uncertainty in a rigorous and repeatable fashion This process has been automated for Butterworth and Chebychev filters of arbitrary order Third order examples are used to illustrate the approach

L_1 methods are used to develop new tuning rules for PID controllers This is achieved using the μ_{L_1} operator which can reasonably be described as a time domain analogue of μ. Used in conjunction with an application of the robust performance theorem, μ_{L_1} is used to find the PID settings that best satisfy a given time domain performance objective

List of Symbols and Acronyms

μ	The Structured Singular Value
μ_{co}	The Convex Estimate of μ
μ_{L_1}	A Time Domain Analogue of μ
k_m	The Multivariable Stability Margin
B_x	A Unit Ball Defined Using the $\| \|_x$ Norm
SISO	Single Input Single Output
MIMO	Multi-Input Multi-Output
LTI	Linear Time Invariant
NL	Non Linear
TV	Time Varying
NLTV	Non Linear Time Varying
BIBO	Bounded Input Bounded Output
RHP	Right Half Plane
LP	Linear Program
SVD	Singular Value Decomposition
CVD	Characteristic Value Decomposition

Chapter 1

Introduction

This thesis is concerned with the structured singular value μ μ is a function which precisely quantifies the effect of particular types of uncertainty on a linear system This introduction gives a background to the research, outlines the problem at hand, the objectives and methodology behind the research, as well as some of the principal limitations of standard μ theory

1.1 Background to the Research

A precise description of uncertainty may seem to be a paradoxical goal at first glance However *exact* certainty about the description of the physical world is never possible The shop is *about* 2 kilometres away A bag of sugar bought in that shop weighs *approximately* 1 kilogramme All measurements are given within a band which is dependent on the accuracy of the instrument involved Thus the shop will be between 1 8 and 2 2 kilometres away The same principle is applied to the models that are developed during a course of systems analysis An example of this is when a student is asked to describe the simple harmonic motion of a pendulum To simplify the system the student is allowed to *approximate* $\mathrm{Sin}(\theta)$ by θ This means that the location of the bob is described by a linear differential equation The student accepts that the model is wrong but that it is *good enough* to allow a tractable yet acceptable analysis of the physical system

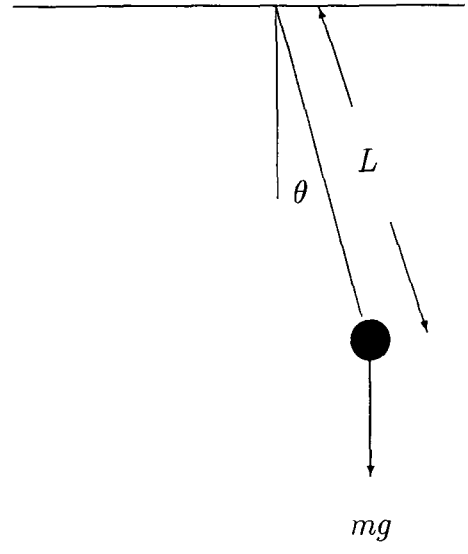


Figure 1 1 The Simple Pendulum

A tradeoff of this nature is common throughout undergraduate control work However, problems arise whenever a student meets a real process, even of the tightly constrained classroom variety A design which performed well on paper and on a computer simulation package, like *SIMULINK*, necessarily suffers from some kind of performance degradation when implemented in practice This degradation is generally due to some error or oversight in the actual system model as opposed to the nominal model that has worked well on paper Examples of this include unmodelled high frequency system dynamics, the effects of sensor noise and the aforementioned linearised approximations of nonlinear physical characteristics

1.1.1 Uncertainty Description

Robust control attempts to overcome these problems by including a full account of the system uncertainty in the model description System uncertainty can be viewed as either **structured** or **unstructured** Unstructured uncertainty occurs when no information is available about the modelling errors for a given physical system except that it is bounded in some fashion The uncertainty is denoted by the operator Δ This Δ acts on the ideal plant P_o in an additive or multiplicative fashion as per Fig. 1.2. Thus, a suitable approach is to bound the norm of the perturbation by a

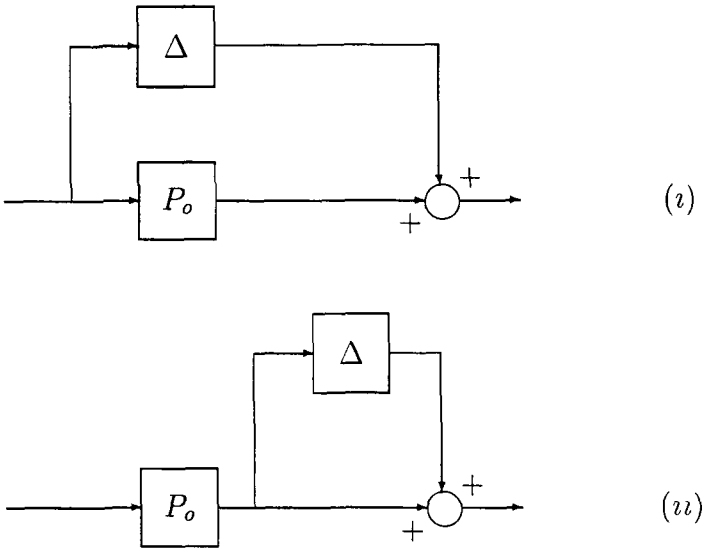


Figure 1 2 (i) Additive and (ii) Multiplicative Unstructured Uncertainty

finite number i e ,

$$\|\Delta\| \leq \alpha$$

This global bounding of uncertainty can be unnecessarily conservative if some knowledge of its structure is available Take, for example a linear filter where the value of a component is allowed to vary by $\pm 10\%$ Perturbing the filter transfer function with a global Δ is a very crude way of representing the effect of this uncertainty

In μ-analysis, the following procedure for structured uncertainty is adopted In any system description there are said to be n sources of uncertainty embedded within the plant Each uncertain element will be denoted by $\Delta_i(j\omega)$ Each $\Delta_i(j\omega)$ may assume any LTI value but is constrained to be less than a certain size This is necessary and actually not as large a restriction as might be anticipated In any process, system components are checked that they are operating within specified tolerances before integration into a larger system

Such dynamical Δ_i's are called **complex** uncertainties It is also useful to consider Δ_i's in the set $\Delta_i \in [-1, +1]$ These Δ_i's are called **real** uncertainties If both types of uncertainty are present then Δ is said to be **mixed** As $\Delta = \{\Delta_1, \quad , \Delta_n\}$ ranges over its permitted values a family of systems will be traced out The system given when each $\Delta_i = 0$ will be termed the **nominal** system Any system where there exists a $\Delta_i \neq 0$ will be termed **perturbed.** A perturbation Δ where all the Δ_i's are

within their predetermined size is **valid** The symbol $\mathcal{D}$ will be used to describe the set (or family) of all validly perturbed systems

1.2 Problem Definition

This research looks at the effect of structured uncertainty on a nominal system This uncertainty creates a family, $\mathcal{D}$, of systems When every element of $\mathcal{D}$ is stable the family is said to be **robustly** stable Clearly, when a controller is added to this nominal system, one obtains a new nominal system In a similar fashion, a performance objective is satisfied robustly if and only if it is met for every element of $\mathcal{D}$ Determining whether a particular family is satisfying a particular performance objective is known as an **analysis** problem Finding a controller so that the corresponding family of systems will perform a suitable objective robustly is known as a **synthesis** problem To illustrate the non trivial nature of the robustness analysis problem the following example is considered

1.2.1 Motivating Example

The EU space agency ARIANE 5 rocket was launched on its maiden voyage from French Guiana on June 4, 1996 Pivoting of exhaust nozzles located at the base of the rocket's two boosters was used to control its trajectory [Radford 96] 37 seconds after lift off these nozzles swivelled abnormally and broke off Ground controllers subsequently detonated the rocket remotely to prevent debris being strewn over a populated area

The official cause of the explosion has not, as yet, been published Speculation at the time centred around a fault in the guidance control system caused by an abnormal shift in payload location within the rocket This would explain the erratic hunting behaviour of the exhaust nozzles and their subsequent failure It may be premature but it is tempting to blame the failure of the launch on a lack of robustness in the system design

1.2.2 A Qualitative Definition of μ

μ theory is the title given to the study of the effect of Linear Time Invariant (**LTI**) structured uncertainty on an LTI system $M(s)$ μ itself can be defined, qualitatively, as follows

Definition 1 For any LTI system with specified levels of structured uncertainty, μ is a measure of how well a stated objective is being achieved, if at all, by that system for all values of frequency

This objective may be robust stability and/or some other performance criteria Some quantitative extension of this definition is required The discussion is limited to stability for the moment The extension to other performance measures will be considered later Clearly, the size of μ will be dependent on the uncertainty set in question

A procedure is required where the nominal system and the uncertainty acting on it can be viewed separately Fortunately this separation process can always be performed with LTI systems [El Ghaoui 91]

1.2.3 The Diagonal Perturbation Formulation

An uncertain system can be viewed as a compound expression of nominal and uncertain elements μ-theory uses a process which separates the nominal from the uncertain elements The Diagonal Perturbation Formulation (**DPF**) is a canonical representation of a system after this separation process has been carried out

An Example

An illustrative example is provided to demonstrate the steps required to convert a system into its equivalent DPF A basic second order system is considered where

$$\frac{y(s)}{u(s)} = \frac{1}{s^2 + 2\zeta\omega_n s + \omega_n^2}$$

$$\equiv \frac{1}{(s+a)(s+b)}$$

noting that

$$a, b \;=\; \omega_n(-\zeta \;\pm\; (\zeta^2 - 1)^{\frac{1}{2}})$$

Uncertainty is introduced by allowing the nominal pole locations a and b to reside anywhere in disks of radius α_a and α_b respectively This type of uncertainty is described as **complex** If the pole locations were only allowed to vary in the direction of their respective real components then the uncertainty would be termed **real** Fig 1 3 shows equivalent representations of this second order system where this complex uncertainty is explicitly taken into account This figure shows how the uncertainty can be "extracted" from the actual system by creating an extra input/output pair for each uncertain parameter Completing this process results in the required block diagonal matrix $\Delta = \mathrm{diag}(\Delta_1, \quad , \Delta_n)$ which acts externally on the nominal plant, $M(s)$ The system is now said to be expressed in its DPF or $M\Delta$ Formulation For this second order system the DPF would be

$$\begin{pmatrix} y \\ r_a \\ r_b \end{pmatrix} \;=\; \frac{1}{(s+a)(s+b)} \begin{pmatrix} 1 & -1 & -(s+a) \\ \alpha_a(s+b) & -\alpha_a(s+b) & 0 \\ \alpha_b & -\alpha_b & -\alpha_a(s+a) \end{pmatrix} \begin{pmatrix} u \\ i_a \\ i_b \end{pmatrix}$$

1.2.4 A Quantitative Definition of μ

It is now possible to consider a proper quantitative definition of μ Consider a system expressed in terms of its DPF as in Fig 1 3 This system will be robustly stable if and only if its nominal system is stable and

$$\det(I + \Delta(j\omega)M(j\omega)) \;\neq\; 0 \qquad \forall\omega \in \mathcal{R}^+,\ \forall\Delta \in \mathcal{D}$$

Thus the following definition of μ (as in, for instance [Packard 93]) is natural

Definition 2

$$\mu_\Delta(M) \;=\; (\inf_\omega\{\inf_{\Delta\in\mathcal{D}}\{\bar{\sigma}(\Delta) \;|\; \det(I + \Delta M) \;=\; 0\}\})^{-1} \qquad (1\,1)$$

where $\bar{\sigma}(\Delta)$ denotes the maximum singular value of Δ Stating the problem in Nyquist terms, $M\Delta$ should not equal minus unity in any vector direction at any frequency

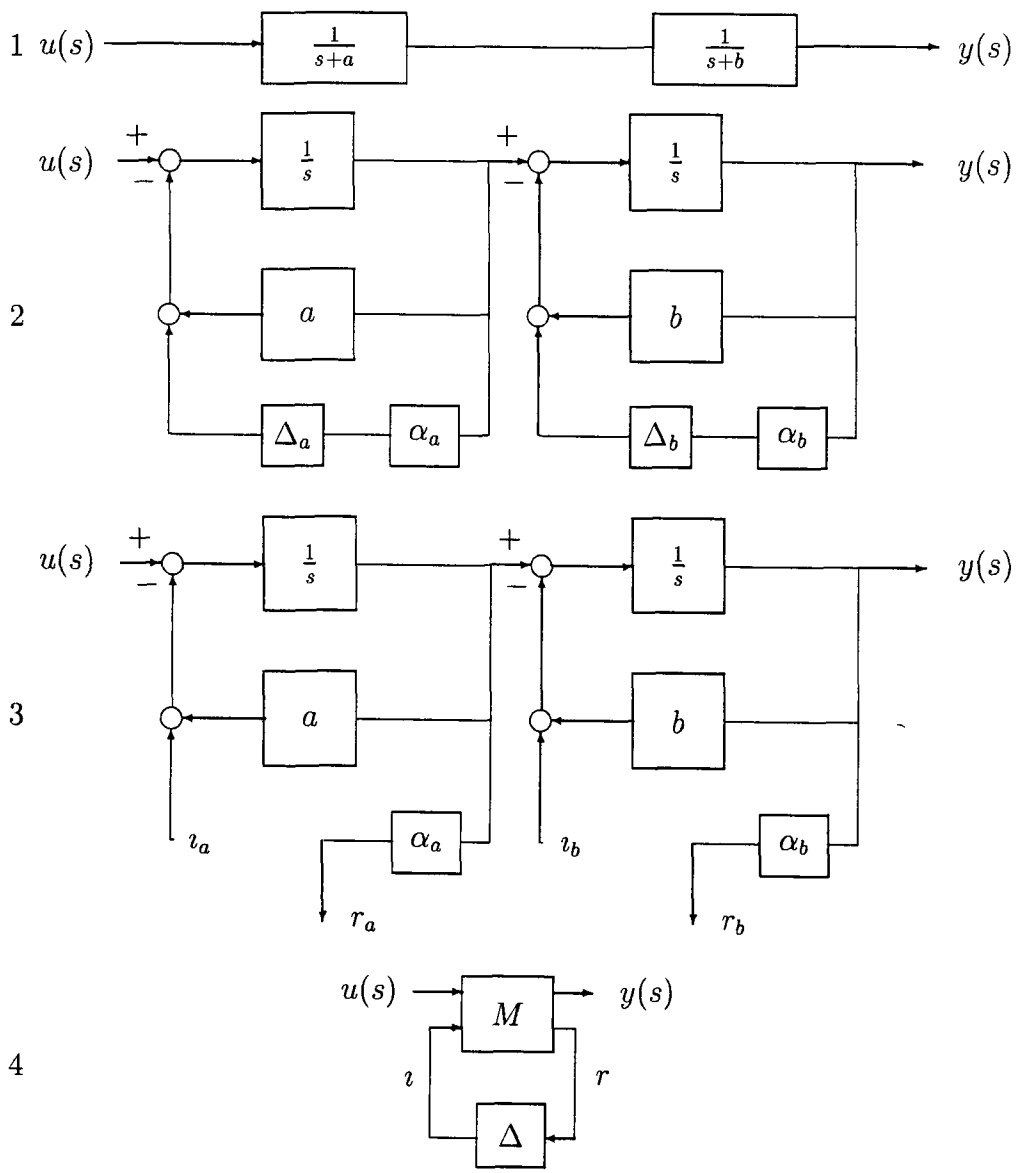


Figure 1 3 Conversion of a typical system to its DPF

Hence, the following quantitative (and appealing) characterisation of μ_Δ is natural for all $\Delta \in \mathcal{D}$ obeying $\bar{\sigma}(\Delta) \leq 1$,

$$\textbf{Robust Stability (RS)} \Leftrightarrow \mu_\Delta(M) < 1$$

Similarly,

$$\textbf{Non Robust Stability (NRS)} \Leftrightarrow \mu_\Delta(M) \geq 1$$

The question of robust stability therefore reduces to computing μ A good deal of the literature uses an alternative terminology to Definition 2 The *multivariable stability margin* can be defined as

$$k_m(M) = \mu_\Delta^{-1}(M)$$

Remark: The Frequency Decoupling Property of μ

The definition of $\mu_\Delta(M)$ indicates that all possible values of frequency must be considered before it can be determined In practice only a relevant subset of frequency is taken into account The overall $\mu_\Delta(M)$ will be the peak value of a frequency dependent function $\mu(\omega)$ Thus $\mu_\Delta(M)$ can be evaluated as a function of frequency in much the same way as the $\| \ \|_\infty$ norm of a system is determined

1.2.5 The Robust Performance Theorem

The discussion so far has been dominated by stability An important feature of μ-theory is its ability to handle performance specifications robustly The procedure for this is fairly straightforward Consider a standard DPF like that of (4) in Fig 1 3 It is possible to close the performance u/y input/output pair with the addition of an extra Δ_f as in Fig 1 4 Thus, $\tilde{\Delta} = \text{diag}(\Delta_f, \Delta_R)$ Clearly, there will be gain and phase information associated with the u/y pair Consideration of the Nyquist interpretation of this information requires that Δ_f may be complex valued The system can now be represented by the $M\tilde{\Delta}$ loop as in Fig 1 4 The **Robust Performance Theorem** [Stein 82] states that a certain specification is satisfied robustly if and only if the augmented $M\tilde{\Delta}$ loop is stable Note how Δ_f is scaled by α in Fig 1 4, so as to better fit performance requirements This is an extremely powerful result It means that suitably defined specifications reduce to a μ test on an augmented version of a

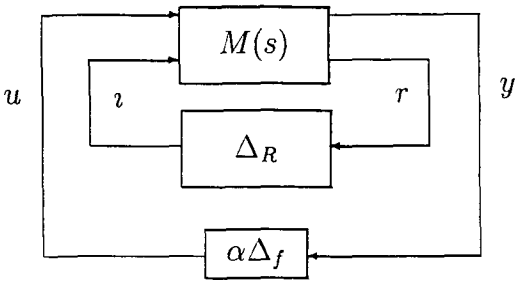


Figure 1 4 DPF with an additional scalar performance parameter

basic DPF There is a straightforward extension to the case of multiple specifications, where there must be a separate complex performance Δ_f for each input output pair

1.3 Objectives of This Work

The primary objective of this work is to efficiently and reliably calculate good estimates of μ Consider Fig 1 4 in the context of eqn (1 1) Depending on the size of α any valid Δ may potentially destabilise the $M\tilde{\Delta}$ loop Therefore *any* destabilising Δ provides an upper bound on $k_m(M)$ The question now arises of how to get from an arbitrary Δ to the Δ which yields the minimiser of eqn (1 1) Unfortunately, this particular question raises the spectre of non-convexity and local extrema

1.3.1 Calculation of μ_{co}, the Need for a Convex Estimate

When attempting to calculate μ a basic problem arises immediately In full generality, the computation of μ is non-convex and NP complete [Demmel 92] Empirically, the required computing time can be observed to grow exponentially with an increase in problem size Therefore, a tractable estimate is required This is achieved by the convexification of the underlying optimisation problem of eqn (1 1) Convex sets are attractive for a number of reasons In particular a geometric formulation of the Hahn-Banach Theorem can be used to determine a convergent algorithm for a certain convex estimate of μ which will be denoted as μ_{co}

μ_{co} provides an upper bound on μ The gap between μ and μ_{co} has been the subject of extensive research [Doyle 82] Evidence for the purely complex case suggests that this convex estimate can be considered "good" For 25000 differently sized problem matrices μ_{co} was at worst 5% greater than a lower bound estimate of μ It should be noted that when mixed uncertainty problems are considered this gap can be made arbitrarily wide However, the literature suggests [Packard 91] that on suitably practically motivated problems the gap is within acceptable levels This claim is examined in this thesis on a well known benchmark problem where μ has been calculated exactly [De Gaston 88]

1.3.2 Determination of the Best Optimisation Strategy for μ

There are two possible ways to obtain bounds on μ The first is a lower bound approach, i e ,

$$\max_{\Delta \in \mathcal{D}} \rho(\Delta M) \;=\; \mu_{\Delta}(M)$$

where $\rho(M)$ denotes the spectral radius of M Recognised ways of solving such a problem include a grid search of suitable valid perturbations or an approach which uses a power iteration to find the spectral radius All lower bound algorithms are compromised by their local character which can yield answers that are arbitrarily far away from the global optimum It should be noted however, that commercial code exists [Balas 94] which uses a power iteration to provide a very good lower bound on μ, particularly in the complex case

The second possible approach is to look at the following upper bound on μ -

$$\mu_{\Delta}(M) \;\leq\; \mu_{co} \;=\; \inf_{valid\ D} \bar{\sigma}(DMD^{-1}) \qquad (1\ 2)$$

In the purely complex case a *valid* D is a matrix which is invertible and commutes with every valid Δ Thus,

$$\mu_{co}(M) \;=\; \mu_{co}(DMD^{-1})$$

since the following determinants are equivalent i e ,

$$\det(I + \Delta(DMD^{-1}))$$

$$= \det(I + D^{-1}\Delta DM)$$

$$= \det(I + \Delta D^{-1} D M)$$

$$= \det(I + \Delta M)$$

The convexity of the problem on the right hand side of eqn (1 2) has motivated the use of the Hahn-Banach Theorem to locate suitable minimising D's An objective of the work is to develop a convergent algorithm that uses this theorem to compute μ_{co} The efficacy of the new approach is also tested against existing commercial code on a broad selection of problem matrices and uncertainty structures Further, the estimate and the algorithm are extended to handle real parametric uncertainty

1.3.3 New Applications for μ

This research endeavours to broaden the potential application base for μ-theory

Filter Performance with Uncertain Components

μ-theory provides a non-probabilistic technique for the assessment of the *worst case* effect of component uncertainty on the performance of a filter circuit This allows guarantees about circuit performance to be made which are very useful from a business/design perspective The problem requires that, firstly, the block diagram for the filter transfer function be recast in its DPF It has been shown that this procedure can be performed on any linear system [El Ghaoui 91] Secondly, it is necessary to map filter performance to an equivalent robust stability question An objective of this part of the work is to develop a procedure which can be applied quite generally to the analysis of any linear filter or analogue circuit with uncertain components, thus demonstrating the broad applicability of μ-theory

The use of a repeatable process like this has major advantages over grid search or Monte Carlo based techniques The cross coupling effect of multiple components are fully taken into account using this novel application of μ

1.4 Limitations of Standard μ Theory

The principal limitation of μ is that it is only defined in terms of LTI perturbations The facility to deal with perturbations that are nonlinear and/or time varying is desirable In this thesis the PID controller has been used as an example where a "μ-like" robustness analysis tool for general perturbations comes in very handy PID performance is known to degrade when plant models are poor Precise information on the best possible PID performance in the face of such perturbations is possible using a μ construct

1.4.1 A μ Theory for Perturbations that are not LTI

Recent results [Dahleh 95] are exploited in this work which allow computation of non-conservative robustness margins in an inexpensive manner These results are loosely grouped under the title of L_1 theory L_1 theory is a time domain approach Time domain specifications are useful in that they are intuitively more easily understood than frequency domain based alternatives

L_1 methods are of particular benefit when the plant model is quite poor Large uncertainty sets are generally required when using poor models to provide a reasonable representation of the real life behaviour of a system In this work these methods have been applied to the tuning of PID controllers The use of a μ-like construct within an L_1 context allows an *exact* assessment of how well a PID control law can fare when faced with a highly uncertain plant, which is subject to non-linear and/or time varying perturbations

Development of New Tuning Rules for PID Controllers

A subsequent objective of this work is to outline the development and implementation of new PID tuning rules using a μ-like construct which will be denoted as μ_{L_1} Rigorous measures of how well time domain performance specifications like worst case tracking error are being satisfied are possible using μ_{L_1}

1.5 Theoretical Framework

One of the strengths of μ is its well laid theoretical foundation In this section a brief overview of the basic properties of μ is provided A much more extensive summary is provided in [Packard 93]

1.5.1 Properties of μ

1 μ is not a norm
$\mu(M)$ of a non-zero matrix M can be zero if there does not exist a valid Δ that will destabilise the loop

2

$$\mu(\alpha M) = |\alpha|\mu(M)$$

This follows directly from the definition of a determinant

3

$$\mu_{\Delta}(M) = 1 \Leftrightarrow \max_{\Delta \in \mathcal{D}} \rho(\Delta M) = 1$$

To see this consider the existence of a valid Δ so that $\rho(\Delta M) > 1$ This means that there exists an x where the vector $\Delta M x$ is a scalar multiple of x and $\|\Delta M x\| > \|x\|$ Scaling the perturbation by β so that the vector $\beta \Delta M x = -x$ yields a singular matrix $(I + \beta \Delta M)$ Thus, $\mu \geq \frac{1}{\beta}$

4

$$\rho(M) \leq \mu(M) \leq \bar{\sigma}(M)$$

The lower bound follows from the previous property where Δ would be an identity matrix The upper bound can be seen by considering the inequality $\bar{\sigma}(M\Delta) \leq \bar{\sigma}(M)\bar{\sigma}(\Delta)$ Let $\bar{\sigma}(M) = \alpha$ Now if $\bar{\sigma}(\Delta) < \frac{1}{\alpha}$ then $(I + M\Delta)$ may be non-singular $\Rightarrow \mu < \alpha = \bar{\sigma}(M)$

5

$$\mu(M) = \mu(DMD^{-1})$$

This was seen earlier

1.5.2 Properties of Mixed μ

The lower bound on μ in property 4 above does not hold when some of the uncertainties are real This is due to the added restrictions imposed on the structure of a valid Δ when real uncertainty is involved The literature [Barmish 90], [Packard 91] has also highlighted the fact that under certain conditions mixed μ can be a discontinuous function of the problem data This has clear implications for any kind of perturbation gridding strategy that may be adopted for the computation of mixed μ Whilst the importance of this result should not be underestimated, steps can be taken in practice to deal with it In particular, [Packard 91] suggests that, from an engineering perspective, this problem is not difficult If a large discontinuity exists then invariably the problem can be diminished by some extra attention being given to the problem/model formulation

1.6 Methodology

There are a number of different ways in which the upper bound of eqn (1 2) can be calculated This thesis uses the numerical range theory of Fan and Tits [Fan 86] to determine it This theory is transparent, self-contained and has a pleasing geometric interpretation Computing this convex estimate, μ_{co}, amounts to determining whether 0 is in a suitable convex set

This work shows that the Hahn-Banach Theorem can be applied to the computation of μ_{co} This theorem says that two disjoint convex sets can be separated by a plane Moreover, location of a suitable separating plane can show that two convex sets are disjoint A suitable norm needs to be chosen to demonstrate that two sets are indeed disjoint This thesis aims to show that use of the 1-norm is the best choice The 1-norm allows the use of Linear Programs (**LP**s), which are attractive due to their widespread usage across a large range of disciplines This combination of use of the Hahn-Banach Theorem allied to a linear program solver makes for a novel way of computing μ_{co}

1.7 Structure of this Thesis

The necessary background material is presented in Chapter 2 The numerical range problem formulation for the computation of μ is discussed Chapter 2 also includes an analysis of the current commercial packages that are available for the computation of μ_{co} The numerical range problem formulation used in this work suggests a number of different algorithmic strategies for μ_{co} computation A number of these possible approaches are also briefly discussed in Chapter 2 However, it is an objective of this work to demonstrate the superiority of one strategy in particular, termed the 1-norm dual approach Chapter 3 is devoted to a comprehensive treatment of this approach along with some necessary extensions for improved performance Chapter 4 is devoted to an extensive validation and performance assessment of the algorithm This assessment falls into two halves Firstly, the new algorithm is compared with existing commercial code on a variety of different problem sizes and uncertainty structures Secondly, the claim that the 1-norm dual approach is indeed the best way to compute μ_{co} is justified

Chapter 5 contains the first application chapter of the work A non-probabilistic technique for the assessment of the *worst case* effect of component uncertainty on filter performance is introduced A general procedure for this process which is applicable for any linear filter is presented Examples of automation of the whole process using Butterworth and Chebychev filters are also included This chapter also demonstrates the key advantage of such a process over a grid search approach by contrasting the different results obtained by the two methods

Chapter 6 outlines the development and implementation of new PID tuning rules Rigorous measures of how well time domain performance specifications like worst case tracking error are being satisfied are demonstrated using an L_1 variation on μ, which throughout this work is denoted by μ_{L_1} The general procedure is given, its features are discussed and some comments about the limitations of such a strategy are also made Two examples of the approach are given Initially a second order system with uncertain damping factor, natural frequency and variable time delay is considered The new rules are shown to compare favourably with existing tuning strategies like those of Ziegler-Nichols The chapter concludes with a similar analysis of a practical, highly nonlinear pH control process The work confirms the belief that

a PID control law is not suitable for such a process

Finally, a summary of the main conclusions from this work and a discussion of possible future research directions are given

1.8 Summary

This chapter has introduced the concept of μ μ-analysis is a respected theory because of its sound theoretical foundation The basic problem of robustness analysis has been discussed Examples have been used to illustrate that it is by no means a solved problem The principal objective of this work is to introduce a novel method for the computation of μ_{co}, the convex estimate of μ, which uses the Hahn-Banach Theorem and a Linear Program solver In addition new applications for μ-theory have been suggested

Chapter 2

Review of Background Material

This chapter reviews the necessary background material for the work that has been carried out. Initially, the Multiform (m-form) numerical range is introduced. Next, the geometric form of the Hahn-Banach Theorem is discussed. This is the principal result that is required to apply the numerical range theory to the question of stability analysis.

The use of the Hahn-Banach Theorem motivates the study of certain optimisation problems which can be solved using either a primal or a dual approach. This chapter looks at 2-norm based primal and dual approaches which allow bounds on μ to be calculated. This thesis argues that a dual approach is a better way to solve the optimisation problem in question. A dual approach requires the use of a Linear Program solver. This motivates a brief discussion of linear programming in general and in particular some of the linear programming issues that impact on the problem at hand.

Commercially available software is available that can determine bounds on μ. A review of the more popular algorithms has been performed. The principal features and failings of existing solutions are highlighted in this chapter.

The chapter concludes with some brief remarks that are relevant to the applications work that has been carried out. While the work on analysis of filter performance in Chapter 5 is pretty much self contained, some comments about L_1 Control Theory

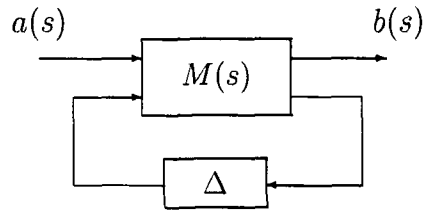


Figure 2 1 An LTI system with uncertain Δ extracted

and the analysis of non-linear systems are included, being pertinent to the PID tuning work of Chapter 6

2.1 Multiform Numerical Range

The fundamental question of robust stability analysis is addressed using the numerical range formulation of Fan *et al* [Fan 91] and [Fan 2 86] Other useful references for this approach are [Holohan 94] and [Fan 88] Consider Fig 2 1 Given an LTI transfer function $M(s)$, the task at hand is to characterise the smallest admissible perturbation Δ that will cause the loop to go unstable Thus for a given perturbation structure $\mathcal{D}$

$$\mu(M)^{-1} = k_m = \inf_{\alpha}\{\alpha \geq 0 \mid \det(I + \alpha\Delta M) = 0 \quad \textbf{for some } \Delta \in \mathcal{D}\}$$

$$= \sup_{\alpha}\{\alpha \geq 0 \mid \det(I + \alpha\Delta M) \neq 0 \quad \forall \Delta \in \mathcal{D}\}$$

Clearly if no such destabilising Δ exists then $\mu(M) = 0$ This leads directly to the following condition for a system being Not Robustly Stable **(NRS)** for fixed α

$$\textbf{NRS} \Leftrightarrow \exists \Delta \in \mathcal{D} \mid \det(I + \alpha\Delta M) = 0$$

Therefore,

$$\textbf{NRS} \Leftrightarrow \exists x \neq 0, \Delta \in \mathcal{D} \mid (I + \alpha\Delta M)x = 0 \qquad (2\,1)$$

Now, $\Delta = \mathtt{diag}\{\Delta_1, \quad, \Delta_m\}$ is a block diagonal matrix Let Δ_i have dimension $q_i \times q_i$ Define the projection matrix P_k, which picks out the kth block of a perturbation structure $\mathcal{K} = \{q_1, \quad, q_m\}$ as

$$P_k = \mathtt{diag}\{0I_{q_1}, \quad, 0I_{q_{k-1}}, I_{q_k}, 0I_{q_{k+1}}, \quad, 0I_{q_m}\}$$

If one thinks in terms of the projection matrix "picking out" a perturbation block of interest it is possible to see how a set of m constraint matrices can be generated

Each one of these matrices must be singular for a system to be NRS Clearly, $\det(I + \alpha\Delta M) = 0$ is equivalent to requiring the existence of a vector x of unit norm such that

$$P_k x + \alpha\Delta P_k M x = 0 \qquad \forall k = 1, \quad , m$$

Taking norms, which introduces a notion of size to the problem, the above statement is true

$$\Leftrightarrow \|P_k x\|_2 \leq \alpha\|\Delta P_k M x\|_2$$

$$\leq \alpha\bar{\sigma}(\Delta)\|P_k M x\|_2$$

Some practical limit must be placed on the size of a perturbation that a system is allowed to see Thus Δ is constrained so that $\bar{\sigma}(\Delta) \leq 1$ Squaring both sides yields,

$$\|P_k x\|_2^2 \leq \alpha^2\|P_k M x\|_2^2$$

By definition, the 2-norm $\|Mx\|_2^2 = x^* M^* M x$ and noting that $P_k = P_k^2$,

$$x^* P_k x \leq \alpha^2 x^* M^* P_k M x$$

$$\Leftrightarrow x^*(P_k - \alpha^2 M^* P_k M)x \leq 0$$

Noting that $P_k = P_k^*$, the following hermitian form can be introduced

$$A_k(\alpha^2) = P_k - \alpha^2 M^* P_k M$$

Thus

$$\mathbf{NRS} \Leftrightarrow x^* A_k(\alpha^2) x \leq 0 \qquad \forall k = 1, \quad , m \tag{2 2}$$

A unit sphere can be defined for a certain norm, i e ,

$$\partial B_2 = \{x \quad \|x\|_2 = 1\}$$

so that ∂B_2 denotes the unit sphere in a 2-norm sense Consider the function f $\mathcal{C}^n \rightarrow \mathcal{R}^m$ where

$$f_k(x) = x^* A_k(\alpha^2) x$$

A more intuitive representation of expr (2 2) can be obtained by consideration of the set $W(\alpha^2)$, defined on a domain of vectors of unit norm, so that

$$W(\alpha^2) = \{w \in \mathcal{R}^m \quad w_k = f_k(x) \qquad x \in \partial B_2\}$$

Thus a set of real m-dimensional w vectors has been generated where a system being not robustly stable will correspond to the existence of w's with non-positive elements Therefore the question of robust stability reduces to

$$\textbf{NRS} \Leftrightarrow \exists x \quad f_k(x) \leq 0 \qquad \forall k = 1, \quad , m$$

$$\Leftrightarrow \exists w \in W(\alpha^2) \qquad w_k \leq 0 \qquad \forall k = 1, \quad , m$$

This is an intuitively appealing geometric framework with which to compute μ All values of α that are less than μ^{-1} will generate vectors where $w_k > 0$ for some k Conversely a value of α that is greater than μ^{-1} will mean that there exists some w with all negative coefficients Figs 2 2 and 2 3 provide an informal two dimensional interpretation

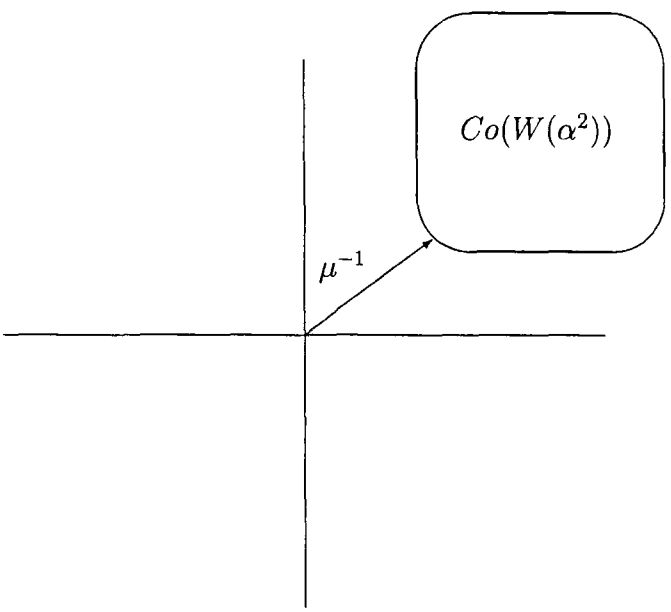


Figure 2 2 Geometric interpretation of the set $Co(W(\alpha^2))$

2.1.1 Complex Block Augmentation

When dealing with complex only uncertainty, the geometric interpretation of Fig 2 3 allows the parameter α to be viewed, rather informally, as a translation scalar on the basic set $Co(W(\alpha^2))$ Fig 2 3 highlights a potential problem Consider the case where the parameter α is increased by a large amount so that it is much greater than

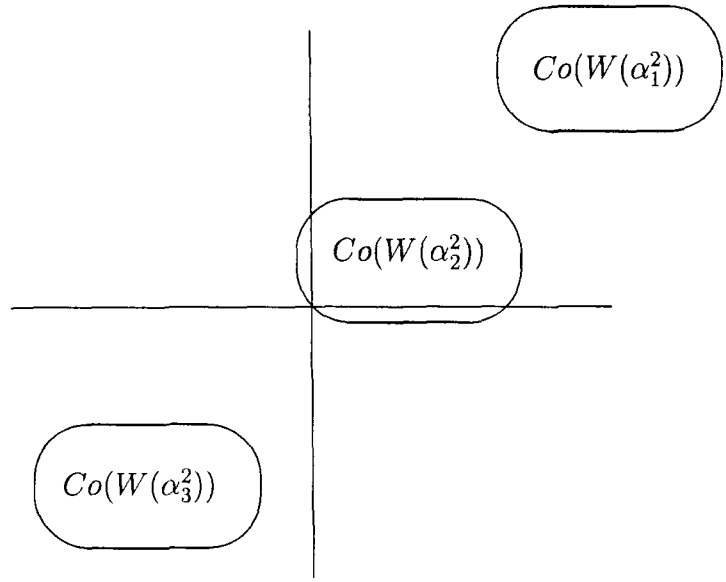


Figure 2 3 Geometric interpretation of the effect of varying the value of α In the figure $\alpha_3 > \alpha_2 = \mu_{co} > \alpha_1$

the stability margin k_m It is possible that the translation effect would push 0 out of the set Thus, an algorithm which determines whether 0 is in the set $Co(W(\alpha^2))$ would cause an incorrect decision about robustness to be made This can be solved by increasing the dimension of the problem through the creation of new variables Expr (2 2) is equivalent to

$$\textbf{NRS} \Leftrightarrow x^* A_k(\alpha^2) x = -\bar{x}_{m+k} x_{m+k}$$

where $\bar{x}_{m+k} x_{m+k}$ corresponds to some positive real number which can be generated from an *extra* complex variable x_{m+k} Thus

$$\textbf{NRS} \Leftrightarrow \hat{x}^* \begin{pmatrix} A_k(\alpha^2) & 0 \\ 0 & P_k \end{pmatrix} \hat{x} = 0 \quad \textbf{where} \quad \hat{x} = \begin{pmatrix} x \\ x_{m+k} \end{pmatrix}$$

$\hat{x}$ is then a vector of dimension $n + m$

Denote the augmented form of $A_k(\alpha^2)$ as $\hat{A}_k(\alpha^2)$ It is clear that

$$\textbf{NRS} \Leftrightarrow \hat{x}^*(\hat{A}_k(\alpha^2))\hat{x} = 0 \qquad \forall k = 1, \quad , m$$

which is an augmented condition for system instability that can deal with any α

Thus, an improved $\widehat{W}(\alpha^2)$ can be considered where

$$\widehat{W}(\alpha^2) = \begin{pmatrix} \hat{w}_1 \\ \\ \hat{w}_m \end{pmatrix} = \begin{pmatrix} \hat{x}^* \hat{A}_1(\alpha^2)\hat{x} \\ \\ \hat{x}^* \hat{A}_m(\alpha^2)\hat{x} \end{pmatrix} \qquad \hat{w} \in \mathcal{R}^m, \hat{x} \in \mathcal{R}^{n+m}$$

Augmentation results in an improved search criteria for robust stability The problem now is to determine whether the null vector is an element of the set $\widehat{W}(\alpha^2)$ i e ,

$$\mathbf{NRS} \Leftrightarrow 0 \in \widehat{W}(\alpha^2)$$

2.1.2 Real Parameters

The discussion so far has been limited to Δ's that are complex valued block perturbations This section shows how numerical range results can be extended to include real valued Δ's When approaching the real case the reader should think in terms of everything that held for the complex case being true with some additional constraints being required to take account of the fact that the perturbation Δ is real valued Again the existence of an $x \neq 0$ is required so that

$$\begin{aligned} & (P_k - \alpha^2 \Delta P_k M)x = 0 \\ \Leftrightarrow \; & (P_k x \quad -\alpha^2 P_k M x) \begin{pmatrix} 1 \\ \Delta_k \end{pmatrix} = 0 \qquad (2\,3) \\ \Leftrightarrow \; & x^*(P_k - \alpha^2 \Delta P_k M)^* = 0 \end{aligned}$$

Noting that $\Delta = \bar{\Delta}$ if the perturbation is real valued, the above is true

$$\begin{aligned} \Leftrightarrow \; & x^*(P_k - \alpha^2 M^* P_k \Delta) = 0 \\ \Leftrightarrow \; & (x^* P_k \quad -\alpha^2 x^* M^* P_k) \begin{pmatrix} I \\ \Delta_k I \end{pmatrix} = 0 \qquad (2\,4) \end{aligned}$$

Evaluation of the simultaneous equations in eqns (2 3) and (2 4) requires that the determinant of the 2×2 submatrix of non-zero entries of the matrix

$$\begin{pmatrix} x^* P_k & -\alpha^2 x^* M^* P_k \\ P_k x & -\alpha^2 P_k M x \end{pmatrix}$$

must equal zero This is true

$$\Leftrightarrow \; x^*(M^*P_k \; - \; P_kM)x \; = \; 0 \tag{2 5}$$

Eqn (2 5) can be seen as the extra constraint required when a perturbation is real Thus for a real perturbation

$$\textbf{NRS} \; \Leftrightarrow \; \exists \hat{x} \neq 0 \; | \; \hat{x}^* \hat{A}_k(\alpha^2)\hat{x} \; = \; 0 \; \textbf{and}$$

$$\hat{x}^* A_k^R \hat{x} \; = \; 0$$

where

$$A_k^R \; = \; \jmath \begin{pmatrix} M^*P_k \; - \; P_kM & 0 \\ 0 & 0 \end{pmatrix}$$

A few comments are appropriate at this stage First the A_k^R terminology has been used to emphasise that the block is independent of α This dictates that the second constraint is associated only with the realness of a perturbation, not its size It is also independent of the augmentation variables $(x_{m+1}, \quad , x_{2m})$

The factor $\jmath$ is required to ensure that A_k^R is a hermitian matrix To see the necessity for this operator consider the 2×2 matrix M where

$$M \; = \begin{pmatrix} a & b \\ c & d \end{pmatrix} \; \Rightarrow M^* \; = \begin{pmatrix} \bar{a} & \bar{c} \\ \bar{b} & \bar{d} \end{pmatrix}$$

$$\Rightarrow M^*P_1 \; - \; P_1M \; = \begin{pmatrix} \bar{a} - a & -b \\ \bar{b} & 0 \end{pmatrix}$$

This matrix needs to be multiplied by $\jmath$ in order to make it hermitian This is also true in general for an $n \times n$ matrix The advantage of this approach is that the question of robust stability again reduces to one of whether $0 \in \widehat{W}$ The extra constraints required for real perturbations means that $\widehat{W}$ will have larger dimension than W For an $n \times n$ matrix subject to n independent, purely real uncertain parameters, it should be noted that an arbitrary w vector will now be of dimension $2n$ i e ,

$$w \in \widehat{W} \; \Rightarrow \; w \; \in \; \mathcal{R}^{2n}$$

2.1.3 Repeated Real Parameters

A similar approach can be used for repeated real parameters This section illustrates how the number of basic constraints increase when a real parameter is repeated Assume that the parameter acts on the ith and jth row and column of the system matrix M As before, one complex $\widehat{A}_k$ matrix is required to represent the constraint on system performance due to α Since the parameter is real

$$P_i x - \alpha^2 \Delta P_i M x = 0$$

and

$$x^* P_i - \alpha^2 \Delta_i x^* M^* P_i = 0$$

must hold simultaneously for some $\hat{x} \neq 0$ However two other conditions must now hold also namely

$$P_j x - \alpha^2 \Delta P_j M x = 0$$

and

$$x^* P_j - \alpha^2 \Delta x^* M^* P_j = 0$$

For these conditions to be true simultaneously it is necessary to have four distinct (augmented) A_k matrices i e ,

$$A_1^{RR} = \begin{pmatrix} P_i M - M^* P_i & 0 \\ 0 & 0 \end{pmatrix}$$

$$A_2^{RR} = \begin{pmatrix} P_i M - M^* P_j & 0 \\ 0 & 0 \end{pmatrix}$$

$$A_3^{RR} = \begin{pmatrix} P_j M - M^* P_i & 0 \\ 0 & 0 \end{pmatrix}$$

$$A_4^{RR} = \begin{pmatrix} P_j M - M^* P_j & 0 \\ 0 & 0 \end{pmatrix}$$

Note that if this parameter had multiplicity 3, i e , it acted on three rows and columns simultaneously, there would need to be 9 A_k^{RR} matrices If the parameter had multiplicity 4 there would have to be 16 matrices and so on Unfortunately there is no guarantee that these matrices are hermitian However given any square A_k and a vector x

$$x^* A_k x = 0$$

$$\Rightarrow \frac{1}{2}x^*(A_k + A_k^*)x = 0 \text{ and}$$

$$\frac{\jmath}{2}x^*(-A_k + A_k^*)x = 0$$

where both the matrices $\frac{1}{2}(A_k + A_k^*)$ and $\frac{\jmath}{2}(-A_k + A_k^*)$ are hermitian To see this, the above can be expanded to yield

$$\frac{1}{2}x^*A_kx + \frac{1}{2}x^*A_k^*x = 0 \text{ and} \tag{2 6}$$

$$-\frac{\jmath}{2}x^*A_kx + \frac{\jmath}{2}x^*A_k^*x = 0 \tag{2 7}$$

Multiplying eqn (2 7) by $\jmath$ and adding it to eqn (2 6) yields the desired requirement Thus the constraint

$$\hat{x}^*A_k^{RR}\hat{x} = 0$$

can also be equivalently represented by the two constraints

$$\hat{x}^*B_k^{RR}\hat{x} = 0 \qquad \text{and}$$

$$\hat{x}^*C_k^{RR}\hat{x} = 0$$

where

$$B_k^{RR} = \frac{1}{2}(A_k^{RR} + A_k^{RR\,*})$$

$$C_k^{RR} = \frac{\jmath}{2}(-A_k^{RR} + A_k^{RR\,*})$$

Therefore 9 hermitian blocks are required to properly represent a real parameter that is repeated once, i e , which has multiplicity 2

$$\textbf{NRS} \Leftrightarrow \exists\hat{x} \neq 0 \mid \hat{x}^*\hat{A}_k(\alpha^2)\hat{x} = 0 \text{ and}$$

$$\hat{x}^*B_k^{RR}\hat{x} = 0 \qquad i = 1, \quad , 2^2 \text{ and}$$

$$\hat{x}^*C_k^{RR}\hat{x} = 0 \qquad i = 1, \quad , 2^2$$

Despite the increase in dimension of the problem, it is noted that the question of robust stability still reduces to the familiar one of whether 0 is in a suitable set $\widehat{W}$

Example: Determining the Dimension of $\widehat{W}$

An example may be useful to illustrate how the dimension of a w vector is adversely affected by repeated real parameters An LTI system is subject to 5 real uncertain parameters Three parameters are independent while one parameter has multiplicity 2 and the fifth has multiplicity 3 Here an arbitrary w will be given by

$$w \in \widehat{W} \Rightarrow w \in \mathcal{R}^{34}$$

In this way there exists one $\widehat{A}_k$ and one A_k^R block for each unrepeated parameter Note that there will always be one $\widehat{A}_k$ block for each parameter For the parameter with multiplicity 2 there will be four B_{ki}^{RR} and four C_{ki}^{RR} blocks representing the constraints required to ensure that the repeated element in Δ is hitting the appropriate rows and columns of M simultaneously Similarly, the parameter with multiplicity 3 will have one $\widehat{A}_k$ block, nine B_{ki}^{RR} blocks and nine C_{ki}^{RR} blocks

Dimension of w for the Full Mixed Uncertainty Problem

The previous example shows how $\widehat{W}$ can quickly become quite complicated Consider the following completely general problem A system with an LTI transfer function is subject to the following LTI uncertain elements

1 n_c uncertain complex blocks

2 n_r uncertain independent real parameters

3 n_{rr} uncertain repeated real parameters each with a (perhaps different) multiplicity given by q_i, $i = [1, \quad , n_{rr}]$

The dimension of a w vector is given by

$$N = n_c + 2n_r + n_{rr} + 2\sum_{i=1}^{n_{rr}} q_i^2$$

2.2 The Hahn-Banach Theorem

The main optimisation result used in this work to estimate μ is the geometric form of the Hahn-Banach Theorem This section briefly presents the necessary concepts for a discussion of this theorem For a more detailed account consult [Lay 82] or [Luenberger 69]

Remark: The Need for a Convex Estimate

Unfortunately the intuitive appeal of the numerical range theory does not mean that the problem is easily solved In fact, in full generality, the problem of whether $0 \in W(\alpha^2)$ is known to be NP hard [Demmel 92] Therefore, the convex hull of $W(\alpha^2)$, which will be denoted as $Co(W(\alpha^2))$, is used instead This is the smallest convex set which contains $W(\alpha^2)$ This will in turn yield a convex estimate for μ which will be denoted by μ_{co} If necessary, this estimate can be improved by the use of domain splitting arguments [De Gaston 88] Clearly,

$$0 \in W(\alpha^2) \;\Rightarrow\; 0 \in Co(W(\alpha^2))$$

Therefore

$$\mathbf{NRS} \;\Rightarrow\; 0 \;\in\; Co(W(\alpha^2))$$

or equivalently

$$0 \notin Co(W(\alpha^2)) \;\Rightarrow \mathbf{RS}$$

The following optimisation strategy will determine whether $0 \in Co(W(\alpha^2))$ for a certain choice of α Introduce $c(\alpha)$ where

$$c(\alpha) \;=\; \min_{w_{co} \in Co(W_{co}(\alpha^2))} \|w_{co}\|$$

Thus,

$$c(\alpha) \;>\; 0 \;\Rightarrow\; \mathbf{RS}$$

2.2.1 Convex Sets

A set $\mathcal{S}$ is **convex** if for each pair of points $x, y \in \mathcal{S}$ then all points of the form

$$z \;=\; \alpha x + \beta y \;\in \mathcal{S} \qquad\qquad \alpha \geq 0,\; \beta \geq 0,\; (\alpha + \beta) = 1$$

Thus for all $x, y \in \mathcal{S}$ the line segment joining x and y is also an element of $\mathcal{S}$ The **convex hull** of a set $\mathcal{S}$, denoted by $Co(\mathcal{S})$, can be defined as the intersection of all the convex sets that contain $\mathcal{S}$ The convex hull of a finite set of points $x_1, \quad , x_{n+1}$ is known as a **polytope** Each of the points $x_1, \quad , x_{n+1}$ is known as a **vertex**

2.2.2 Separating Hyperplanes

In this work, the Hahn-Banach Theorem is used to verify convex hull membership Before doing this a key property of a convex set is introduced Our interest lies in formulating hyperplanes as linear functionals The simplest example of a hyperplane is a line in standard cartesian coordinates Consider

$$H = \{(x, y) \in \mathcal{R}^2 \quad x + 2y = 3\}$$

This can be recast as a linear functional i e , $f \quad \mathcal{R}^2 \to \mathcal{R}$ where $f(x, y) = x + 2y$ and the hyperplane is $H = \{f \quad c\}$ In this case

$$f = [1\ 2] \qquad c = [3]$$

Definition The hyperplane $H = \{f \quad c\}$ separates the sets A, B if either

$$f(a) \leq c \quad \forall a \in A \qquad \textbf{and} \qquad f(b) \geq c \quad \forall b \in B$$

or

$$f(a) \geq c \quad \forall a \in A \qquad \textbf{and} \qquad f(b) \leq c \quad \forall b \in B$$

Removing the possibility for equality to c imposes a strict separation condition on the two sets A, B

Definition Consider $x \in \mathcal{S}$ such that x is on the boundary of $\mathcal{S}$ Let H be a hyperplane such that $\mathcal{S} \cap H = \{x\}$ H is then said to be a *support hyperplane* of the set $\mathcal{S}$

The following facts about separating and support hyperplanes are standard For proof, the reader may consult, for instance, [Lay 82] Denote A° as the interior of a convex set A

Properties of Separating/Support Hyperplanes

Suppose that A and B are convex sets

1 If $A^\circ \neq \emptyset$ and $B \cap A^\circ = \emptyset$ then there exists a hyperplane which separates A and B

2 If A is compact and B is closed then there exists a hyperplane which strictly separates A and B if and only if $A \cap B = \emptyset$

3 If x is a boundary point of A then there exists at least one hyperplane supporting A at the point x

4 If for every boundary point of the set A there exists a support hyperplane then the set A is convex

2.2.3 Primal/Dual Approaches

For any linear functional defined on a compact convex set $\mathcal{S}$, there exists $\bar{x}, \hat{x} \in \mathcal{S}$ such that

$$f(\bar{x}) = \min_{x \in \mathcal{S}} f(x)$$
$$f(\hat{x}) = \max_{x \in \mathcal{S}} f(x)$$

The existence of such extrema for convex sets will be used to determine whether $0 \in Co(W(\alpha^2))$ For any $\alpha > 0$

$$\mu_{co}^{-1} > \alpha \Rightarrow 0 \notin Co(W(\alpha^2))$$

$$\Leftrightarrow c(\alpha) = \min_{w_{co} \in Co(W(\alpha^2))} \|w_{co}\| > 0$$

$c(\alpha)$ can be ascertained by tackling this minimisation problem directly This is known as a primal approach

The term duality crops up regularly in the literature In its most general sense it refers to two similar theories that are related, in an "inverse" or "mirror-image" way The dual space of a vector space X will be denoted by X^* Duality finds application in this work by considering the dual space of linear functionals on a standard vector space

A dual approach, in the context of the problem at hand, will mean the development of a suitable convergent series of linear functionals which establishes whether there exists a suitable f such that $c(\alpha) > 0$ This dual approach requires the Hahn-Banach Theorem

2.2.4 Statement of the Hahn-Banach Theorem

A survey of the literature will reveal several different statements of the Hahn-Banach Theorem This is a version from [Lay 82] which is known as the *geometric form* of the Hahn-Banach Theorem

Theorem 2.1 *Suppose that A and B are closed convex sets of $\mathcal{R}^n$ such that $A \cap B = \emptyset$ If B is compact then there exists a hyperplane H that strictly separates A and B*

Consider a hyperplane H defined by a linear functional f where

$$H = \{f \quad c\}$$

The Hahn-Banach Theorem states that if $A \cap B = \emptyset$ then a linear functional f exists such that

$$< f, a > < < f, b > \qquad \forall a \in A \quad \forall b \in B$$

A dual algorithm for the computation of μ_{co} searches for a suitable separating plane f In this case B will be a closed ball of vanishingly small radius centred at the origin Thus, the existence of a separating f determines whether $0 \in Co(W(\alpha^2))$

A two dimensional geometric interpretation of this theorem can be seen in Fig 2 4 Consider the case where the set B is a sphere whose radius is unity i e ,

$$B = \{x \quad \|x\|_2 \leq 1\}$$

Consider the hyperplane H_1 where

$$H_1 = \{f \quad c\} \qquad f = [0, 1] \qquad c = [1]$$

It is clear from the figure that if the distance from the set A to the origin is greater than unity then $A \cap B = \emptyset$ This means that A is separated from the open ball B by H_1

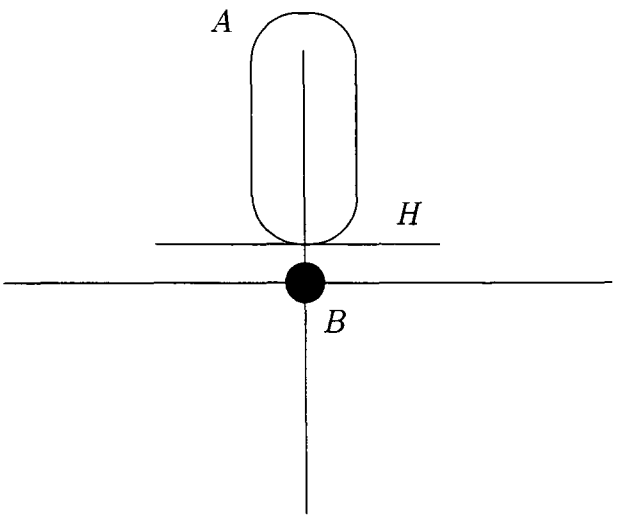


Figure 2 4 Two dimensional geometric interpretation of the Hahn-Banach Theorem

2.3 A Primal Algorithm for the Computation of μ_{co}

In this chapter the 2-norm will be used to illustrate examples of primal and dual methods for the computation of μ_{co} The primal problem is to find

$$c_2(\alpha) = \min_{w_{co} \in Co(W(\alpha^2))} \|w_{co}\|_2$$

A sequence $\{w^{(n)}\}$ is generated according to the following algorithm

1 Find an initial (hopefully good) $w^{(1)} \in Co(W(\alpha^2))$

2 Find the steepest descent direction, $v^{(n)}$, from the current best $w^{(n)}$

3 Do a line search to locate w^{n+1}, the minimum of $\|\alpha w^{(n)} + (1-\alpha)v^{(n)}\|_2$

4 Increment n Go to step 2

2.3.1 Locating Descent Directions

Consider an arbitrary $v \in W(\alpha^2)$ Let $w^{(n)}$ equal the best vector that has been found thus far, i e , the vector in $W(\alpha^2)$ which has smallest norm It can be shown that v

is a descent direction from $w^{(n)}$

$$\Leftrightarrow < w^{(n)}, v > \; < \; \|w^{(n)}\|_2^2$$

which is true

$$\Leftrightarrow \; \sum_{k=1}^{m} w_k^{(n)}(x^* A_k x) \; < \; \|w^{(n)}\|_2^2$$

for some candidate vector x of unit norm Thus, for this candidate x vector

$$x^*(\sum_{k=1}^{m} w_k^{(n)} A_k)x \; < \; \|w^{(n)}\|_2^2$$

Let $\lambda_{min}(\,)$ denote " Minimum Eigenvalue of " Descent directions from $w^{(n)}$ can only exist if

$$\lambda_{min}(\sum_{k=1}^{m} w_k^{(n)} A_k) \; < \; \|w^{(n)}\|_2^2$$

Further, the eigenvector(s) which correspond to λ_{min} determine $v^{(n)}$, the vector of steepest descent

2.3.2 Line Search

This subsection discusses Step 3 in the primal algorithm Determination of $w^{(n+1)}$ is achieved by analysis of the line segment

$$(1-\epsilon)w^{(n)} \; + \; \epsilon v^{(n)}$$

where $\epsilon \in [0,1]$ Viewing this line as a continuous function of the parameter ϵ, i e ,

$$f(\epsilon) \; = \; \|(1-\epsilon)w^{(n)} \; + \; \epsilon v^{(n)}\|_2^2$$

and noting that

$$\frac{df}{d\epsilon} \; = \; -2(1-\epsilon)\|w^{(n)}\|_2^2 \; + \; 2\epsilon\|v^{(n)}\|_2^2 \; + \; (2-4\epsilon) < w^{(n)}, v^{(n)} >$$

it is clear that the required minimiser will correspond to where $\frac{df}{d\epsilon} \; = \; 0$ This will occur when

$$\epsilon \; = \; \frac{\|w^{(n)}\|_2^2 \; - \; < w^{(n)}, v^{(n)} >}{\|w^{(n)} - v^{(n)}\|_2^2}$$

The algorithm outlined in this section is computationally inexpensive and will continue as long as descent directions are being located This algorithm has been implemented in *MATLAB* A full analysis of its performance is given in Chapter 4 of this thesis

2.4 A Dual Algorithm for the Computation of μ_{co}

The previous algorithm will (eventually) locate $c_2(\alpha)$ For any positive ϵ

$$RS \Leftarrow Co(W(\alpha^2)) \cap \epsilon B_2^\circ = \emptyset \tag{2 8}$$

$$\Leftrightarrow \epsilon < c_2(\alpha)$$

The Hahn-Banach Theorem says that eqn (2 8) holds if and only if there is a separating hyperplane $d \neq 0$ for which

$$< d, \epsilon b > \leq < d, w_{co} > \qquad \forall w_{co} \in Co(W(\alpha^2)), \quad \forall b \in B_2$$

$$\Leftrightarrow \epsilon \|d\|_2 \leq < d, w_{co} > \qquad \forall w_{co} \in Co(W(\alpha^2))$$

$$\Leftrightarrow \epsilon \|d\|_2 \leq < d, w > \qquad \forall w \in W(\alpha^2)$$

$$\Leftrightarrow \epsilon \leq < \frac{d}{\|d\|_2}, w >$$

$$\Leftrightarrow \lambda_{min}(\sum_{k=1}^{m} \frac{d_k}{\|d\|_2} A_k) \geq \epsilon$$

As this condition holds when ϵ is made vanishingly small

$$\Leftrightarrow \lambda_{min}(\sum_{k=1}^{m} \frac{d_k}{\|d\|_2} A_k) > 0 \tag{2 9}$$

The dual approach attempts to find the linear functional (hyperplane) d that maximises λ_{min} in the inequality of (2 9) An algorithm is thus required which generates an appropriate sequence of functionals $\{d^{(n)}\}$ Consider

$$d^{(n+1)} = d^{(n)} + \delta$$

the requirement that δ increases λ_{min} means that

$$\lambda_{min}(\sum_{k=1}^{m} \frac{d_k^{(n)} + \delta_k}{\|d^{(n)} + \delta_k\|_2} A_k) \geq \lambda_{min}(\sum_{k=1}^{m} \frac{d_k^{(n)}}{\|d^{(n)}\|_2} A_k) \tag{2 10}$$

It should be noted that this must be true for all minimum eigenvalues even if λ_{min} has a multiplicity that is greater than one Consider a finite list of w vectors Expr (2 10) implies that for some small positive scalar β

$$< \frac{d^{(n)} + \beta\delta}{\|d^{(n)} + \beta\delta\|_2}, w^{(i)} > \geq < \frac{d_k^{(n)}}{\|d^{(n)}\|_2}, w^{(i)} >$$

where $i = [1, \quad, p]$ and p is the multiplicity of $\lambda_{min}(\sum_{k=1}^{m} \frac{d_k^{(n)}}{\|d^{(n)}\|_2} A_k)$ Each eigenvector $x^{(i)}$ gives a corresponding $w^{(i)}$ It can be shown [Holohan 97] that this is true

$$\Leftrightarrow < \delta, w^{(i)} > \geq 0 \qquad \forall i = 1, \quad, p \qquad \textbf{and} \qquad (2\ 11)$$

$$< \delta, d^{(n)} > = 0$$

where without loss of generality δ can be rescaled so as that $\|\delta\| = 1$ The orthogonality requirement comes from the fact that any component of δ in the direction of $d^{(n)}$ will have no effect on the objective at hand

The need for a Linear Program (LP) Solver

The largest possible increase in λ_{min} will be attained when the LHS of the inequality in (2 11) is maximised The steepest possible ascent direction for λ_{min} can thus be found using an LP solver The problem at hand is to find the δ that

$$\textbf{Maximises} \quad \tau \quad \textbf{Subject To}$$

$$< \delta, w^{(i)} > \geq \tau \qquad i = 1, \quad, p$$

and

$$< \delta, d > = 0$$

Outline of the Algorithm

1 Determine a (hopefully good) starting linear functional (hyperplane) $d^{(1)}$ and corresponding $\lambda_{min}(\sum_k d_k^{(1)} A_k)$

2 Using an LP solver, determine an ascent direction, δ, from $d^{(n)}$

3 Use a line search in the direction δ to determine the $d^{(n+1)}$ which will yield the maximum possible increase in λ_{min}

4 Increment n Go to Step 2

The algorithm will continue as long as a positive τ is returned from the LP solver in Step 2

Limitations of this approach

The principal drawback of such an approach is that its performance depends on the quality of the initial hyperplane that is used Another drawback is the computational expense of the algorithm's requirements for a Linear Program *and* a line search at each step Despite this, a 2-norm dual approach can be useful because of the improvement in λ_{min} that is possible at each iteration The level of improvement per iteration is radically better than with a primal approach Bearing in mind that termination occurs when a positive λ_{min} has been found, a 2-norm dual approach may be competitive at lower accuracy levels

An analysis of the performance of such an algorithm is given in Chapter 4

2.5 Linear Programming

Any dual approach will require the use of Linear Programming (**LP**) software The algorithm in the previous section and the algorithm presented in the next chapter motivates a brief discussion of the computational issues involved in linear programming The literature on linear programming is vast No attempt is made in this thesis to provide a comprehensive treatment of the problem However, the solution of a linear program is central to the approach adopted here Brief comments which are relevant to the problem at hand are necessary Initially the basic problem is briefly reviewed This section considers the Simplex Method, which is the principal linear program solver used throughout this project The emphasis in this discussion is on the specific problems that the method has with the data that are typically given to the Simplex LP solver

Finally, some brief comments are made about a different Interior Point approach to Linear Program solution The literature [Astfalk 92], [Beran 95] suggests that interior point methods are more efficient than the Simplex Method, particularly on large problems

2.5.1 Statement of the Problem

The linear programming problem is to maximise the objective function

$$\lambda_{LP} = C^T x$$

subject to the constraints

$$\begin{aligned} A_1 x &\leq B_1 \quad b_i \geq 0 \quad i = 1, \quad , m_1 \\ A_2 x &\geq B_2 \quad b_i \geq 0 \quad i = m_1 + 1, \quad , m_2 \\ A_3 x &= B_3 \quad b_i \geq 0 \quad i = m_2 + 1, \quad , m \end{aligned}$$

and also subject to the primary constraints

$$x_1 \geq 0, \; x_2 \geq 0, \quad , \; x_n \geq 0,$$

Any linear program can be expressed in this form Any solution x that satisfies the constraints is known as a feasible solution The feasible solution that maximises the objective function is known as the *optimal feasible solution* The existence of an optimal feasible solution is dependent on two things (i) Feasible solutions must exist (ii) The constraint set must prevent any variable assuming such values that will cause the objective to be unbounded

2.5.2 The Simplex Method

First published by Dantzig [Dantzig 63] in 1948, the Simplex Method provides a simple and, with a few amendments, most notably one by Bland [Bland 77], deterministic way of calculating the optimal feasible vector for any given linear program The steps required in a typical simplex method can be briefly outlined as follows A good general reference for this work is [Spivey 70]

1 **Construct the Basic Tableau** A basic tableau is of the form

$$\begin{bmatrix} \lambda_{LP} & C^T \\ B & -A \end{bmatrix}$$

Initially λ_{LP} will have the value zero

2 **Generate Slack Variables** Inequalities must be turned into equalities by the addition of slack variables $y_1, \quad , y_{m1+m2}$ Initially the non-slack variables $x_1, \quad , x_n$ will have a value of zero The slack variable y_i will initially have the b_i value corresponding to that constraint

3 **Locate the Pivot Column** Looking at the objective function, the basic variable whose coefficient is most positive, i e , that which will most quickly raise λ_{LP} from zero, is selected as the pivot column

4 **Locate the Pivot Element** Assuming that an optimal solution exists and to keep the equations consistent, one raises λ_{LP} by reducing the value of a basic variable Simplex determines the first basic variable that will go non-positive by using the *choice rule*

$$y_h = \frac{b_h}{a_{hk}} = \min_i \{\frac{b_i}{a_{ik}}\}$$

where k corresponds to the pivot column

5 **Swap a Basic/Non-Basic Variable Pair.** The simplex method raises the LP cost by swapping a basic and non-basic variable Standard row/column matrix operations will allow an update of the tableau so that it remains in **normal** form This means that a basic variable will have a coefficient of 1 in the *only* row that it appears in

6 **Repeat Steps 3-5** Do so until a pivot column that will increase the LP cost can no longer be found

7 **Read the Optimal Solution** The optimal solution vector can be read directly from the first column of the tableau

2.5.3 Computer Implementation

The algorithm implemented in this project is based on the work of Kuenzi, Tzschach and Zehnder which is presented in the popular Numerical Recipes suite of software [Flannery 91] It has been implemented, (without the GOTO loops!), in *MATLAB* When only $\leq$ constraints exist it is possible to read a basic feasible solution directly from the tableau This algorithm ascertains whether a basic solution exists

by constructing an (initially negative) auxiliary objective function from the $\geq$ and $=$ constraints If this auxiliary objective can be increased to zero then a basic feasible solution is known to exist This approach also simplifies the row/column operations necessary when handling different types of constraints

The *MATLAB* implementation handles degenerate problems Degeneracy is observed when ties occur during the execution of the choice rule This can be a serious problem resulting in an infinite number of swaps between basic and non-basic variables without any improvement in the LP cost or termination of the algorithm The algorithm that has been used is equipped with an anti-cycling routine due to Bland [Bland 77] which assures finiteness Bland's method bases the decision about which row/column to swap on an index of basic variables The basic variable with the lowest index on the list is always used and thus the number of swaps that occur will always be finite

2.5.4 Interior Point Methods

Interior point methods have been presented by many authors, including Karmarkar [Karmarkar 84], Boyd *et al* [Boyd 94] and Astfalk *et al* [Astfalk 92], as a means of solving linear programs efficiently The approach converts a standard constrained minimisation problem to a corresponding family of unconstrained minimisation problems The method uses a barrier function $BF(x, \gamma)$

$$BF(x, \gamma) = C^T x - \gamma \sum_{j=1}^{n} log(x_j) \qquad (2\ 12)$$

where γ is a positive parameter Any solution of the unconstrained optimisation problem will be in the interior of the constrained solution space Let $\hat{x}(\gamma)$ be the minimiser of eqn (2 12) As γ approaches zero, the minimiser $\hat{x}(\gamma)$ approaches the optimal solution of the constrained minimisation problem A typical algorithm, [Astfalk 92], which locates a minimum in such a way is

1 Choose $\gamma_0 > 0$ Iteration counter k is zero

2 Find $\hat{x}_k$ which is a minimiser of (2 12) This is achieved using a standard Newton-Raphson i e ,

$$\frac{\partial(BF(x, \gamma))}{\partial x_j} = \frac{\partial C^T(x)}{\partial x_j} - \frac{\gamma}{x_j}$$

3 if $\gamma_k < \epsilon$ **STOP**
ELSE $\gamma_{k+1} = \gamma_k * 995$
$k = k + 1$
GOTO Step 2

The attractiveness of an interior point method lies in its computational speed The solution of a Newton-Raphson at each iteration is inexpensive In Chapter 4 interior point code developed by Boyd *et al* [Boyd 94], is compared with the Numerical Recipes Simplex routine

2.6 Overview of Some Commercially Available Software

This section examines some of the commercial μ software which has been available for some time now

2.6.1 MFD Toolbox

Early attempts at computing μ allowed only complex uncertainty and limited uncertainty structures to be used One of these, which is available in the *MATLAB* Multivariable Frequency Domain **(MFD)** Toolbox is due to Ford *et al* [Ford 90] The MFD Toolbox code attempts to find the $\widehat{D}$ which will minimise the upper bound on μ_{co}, i e ,

$$\bar{\sigma}(\widehat{D}M\widehat{D}^{-1}) = \inf_{D\in\mathcal{D}} \bar{\sigma}(DMD^{-1}) = \mu_{co} \qquad (2\ 13)$$

This code uses standard first derivative gradient based and second derivative Hessian based operations to iteratively improve D Such a "hill climbing" approach suffers from an unavoidable reduction in performance during its latter stages Example problems can be found where this reduction in performance can occur at a significant distance away from μ_{co} In addition, real/mixed uncertainty problems are not catered for

2.6.2 μTools

The μTools Toolbox for *MATLAB* [Balas 94] has been recently introduced It offers significant improvements on the MFD Toolbox code It offers faster upper bound code for μ_{co} based on an improved form of eqn (2 13)

$$\bar{\sigma}\left[(I+G^2)^{-\frac{1}{4}}\frac{1}{\alpha_{\mu Tools}}(DMD^{-1} - jG)(I+G^2)^{-\frac{1}{4}}\right] \tag{2 14}$$

The code uses a dual process to determine a good D For a given perturbation structure [Young 95]

$$\mu_{co}(M) < \frac{1}{\alpha_{\mu Tools}} \Leftarrow \bar{\sigma}(\,) < 1$$

The G matrix contains the extra constraints required for real perturbations The bound is exactly equivalent to the real/mixed bounds that are developed from the numerical range theory An iterative (dual) process is required to determine D Termination is based on when the improvement per iteration falls below a certain (prescribed) level This can occur, particularly in the real case, at a significant distance away from μ_{co} μTools largely overcomes this problem in the purely complex case by using a power iteration algorithm which closes in on the lower bound for μ In the complex case this power iteration yields a *very* accurate bound for μ However its performance deteriorates when mixed uncertainty problems are studied

Power Iterations

A power iteration is a procedure which iteratively computes the eigenvalues and eigenvectors of a matrix Consider an arbitrary $x^{(0)}$ with $\|x^0\|_2 = 1$ The following is an illustration of a power method

1 $z^{(n)} = Mx^{(n-1)}$

2 $x^{(n)} = \frac{z^{(n)}}{\|z^{(n)}\|_2}$

3 $\Lambda^{(n)} = x^{*(n)}Mx^{(n)}$

4 Increment n Go to Step 1

The principal objection to power methods is that they are local The choice of $x^{(0)}$ has a bearing on the quality of result obtained However, they are particularly computationally inexpensive

Remark: Comparison of $\alpha_{\mu Tools}$ and Numerical Range Seed α

Eqn (2 14) shows that in general one can think of the following relationship between the corresponding performance measures in the Multiform Numerical Range (α) and μTools ($\alpha_{\mu Tools}$) formulations

$$\alpha_{\mu Tools} = (\alpha)^{\frac{1}{2}}$$

It should be noted that the μTools package does not support the possibility that the performance parameter, $\alpha_{\mu Tools}$, may in some applications not be required to hit *all* the uncertain blocks in a given system A clear practical motivation for this requirement will be seen in the filter analysis problem of Chapter 5 Indeed, μTools cannot be used in its present form to determine worst case filter performance

2.7 L_1 Theory

In this section stability and performance robustness measures are considered for perturbation structures which are no longer, of necessity, Linear and Time Invariant (LTI) [Dahleh 95] is accepted as the reference textbook in this area

l_1 signals are those whose 1-norm, defined by

$$\|y(t)\|_1 = \int_{t=0}^{\infty} | y(t) | \, dt$$

is finite In the sampled data case l_1 signals are those where

$$\|y(k)\|_1 = \sum_{k=0}^{\infty} | y(k) | < \infty$$

l_∞ signals are those whose ∞-norm, defined by

$$\|y(t)\|_\infty = \sup_{-\infty \leq t \leq \infty} |y(t)|$$

is finite In the discrete time case,

$$\|y(k)\|_\infty = \sup_k |y(k)| < \infty$$

The L_1 system norm is induced by l_∞ signals A system is said to be L_1 stable if and only if all bounded input signal vectors (in the infinity norm) will yield a bounded output vector for all time L_1 control is a time domain approach A distinct advantage of a time domain approach is that one can work with objective functions that are intuitively appealing to the engineer For instance tracking error has a much more immediate meaning in terms of output quality than the energy functions which are minimised with a H_∞ approach

As before, a set of LTI perturbations will be denoted by $\mathcal{D}$ Now let $\mathcal{D}_{LTV}$ denote a set of linear but time varying perturbations Further, let $\mathcal{D}_{NLTV}$ denote a set of non linear time varying perturbations A typical general perturbation set might contain perturbations of the form

$$\Delta_{ex} = \mathbf{diag}\,[\Delta_1, \Delta_2, \Delta_3], \qquad \Delta_1 \in \mathcal{D},\ \Delta_2 \in \mathcal{D}_{LTV},\ \Delta_3 \in \mathcal{D}_{NLTV}$$

where the gain of Δ_{ex} will be bounded by unity in the infinity norm sense Let this general perturbation set be denoted by $\mathcal{D}_{ex}$

2.7.1 A μ Construct for General Perturbation Sets

μ-theory can be extended to a general Δ_{ex} Note that [Dahleh 95] refers to this value as a *Structured Norm*, and denotes it as $SN_{\Delta_{ex},\infty}(M)$ The terminology $\mu_{L_1}(M)$ is prefered here to emphasise the fact that it is not a norm Also the Structured Norm $SN(M)$ is defined for any bounded signal set In this thesis, the μ_{L_1} notation refers exclusively to signals that are bounded in an infinity norm sense Thus

$$\mu_{L_1}(M) = \frac{1}{\inf_{\Delta_{ex}}\{\|\Delta\|_1 \mid \det(I - M\Delta) = 0\}}$$

As with standard μ theory, if there is no Δ that will destabilise M then $\mu_{L_1}(M) = 0$ Therefore the use of the μ_{L_1} notation can be considered reasonable Consider the following properties of $\mu_{L_1}(M)$ which follow directly from its definition

Properties of $\mu_{L_1}(M)$

1 $\mu_{L_1}(M) \leq \|M\|_1$

2 $\mu_{L_1}(M) \leq \inf_{\text{valid D}} \|DMD^{-1}\|_1$

3 $\mu_{L_1}(M) = \mu_{L_1}(DMD^{-1})$

4 $|\det(I - M\Delta)| > 0 \quad \forall \Delta \in B_{\Delta_{ex},\infty} \Leftrightarrow \mu_{L_1}(M) < 1$

In Property (2), a valid D will be determined by the perturbation set in question Property (4) is, in essence, a restatement of the small gain theorem for structured, non linear and time varying perturbations

Evaluating $\mu_{L_1}(M)$

$\mu_{L_1}(M)$ is computed as follows It can be taken that M is square and has dimension n Thus, let

$$M(s) = \begin{pmatrix} m_{11}(s) & , & , & m_{1n}(s) \\ & & & \\ m_{n1}(s) & , & , & m_{nn}(s) \end{pmatrix}$$

Let $m_{ij}(t)$ denote the inverse Laplace transform of $m_{ij}(s)$, i e , the corresponding impulse response Thus, $\|m_{ij}(t)\|_1$ denotes the integral of the absolute value of $m_{ij}(t)$ so that

$$\|m_{ij}(t)\|_1 = \int_0^\infty |\, m_{ij}(t)\,| \; dt$$

In this way one can calculate the 1-norm of the impulse response of each element of M Define $\widehat{M}$ as

$$\widehat{M} = \begin{pmatrix} \|m_{11}\|_1 & , & , & \|m_{1n}\|_1 \\ & & & \\ \|m_{n1}\|_1 & , & , & \|m_{nn}\|_1 \end{pmatrix}$$

The spectral radius of $\widehat{M}$, $\rho(\widehat{M})$ has a number of interesting properties Proof of the following theorem can be found in [Dahleh 95]

Theorem 2.2 *Consider a matrix M acted on by a general unity gain bounded (in the infinity norm sense) perturbation set $\mathcal{D}_{ex}$ Let $\rho(\widehat{M})$ denote the spectral radius*

of matrix $\widehat{M}$, consisting of the 1-norm of each element of M. Then the following statements are equivalent.

1. $\rho(\widehat{M}) \leq 1$
2. The system of inequalities
$$x < \widehat{M}x \text{ and}$$
$$x \geq 0$$
has no solutions.
3. $\inf_{\text{valid D}} \|DMD^{-1}\|_1 \leq 1$
4. $\mu_{L_1}(M) \leq 1$

Moreover it can be shown that $\mu_{L_1}(M) = \rho(\widehat{M})$

This is an extremely useful and powerful result. The theorem shows that $\rho(\widehat{M}) \leq 1$ provides necessary and sufficient conditions for stability and performance robustness. It is also a computationally inexpensive function to calculate. It is much cheaper for example than computing $\mu_{co}(M)$ when a perturbation Δ is purely LTI. Extensive use of this result is made in the PID tuning work of Chapter 6.

2.8 Non-Linear Systems

The L_1 theory is of particular benefit when the system model is quite poor. A common example of this is when a non-linear system is approximated by an LTI system for the purposes of design. LTI models are valid locally in the region of an *equlibrium point.* When a gain scheduled type approach is adopted several LTI systems may be required in different regions of operation to better describe non-linear system behaviour. It is inevitable that problems will occur with the faithfulness of linear approximations, particularly at the transition points between regions. The L_1 results of the previous section can be of benefit here. This section describes the steps required to linearise a model about an equilibrium point. There are many useful references for this work, [Kaplan 95] being a particularly well written example.

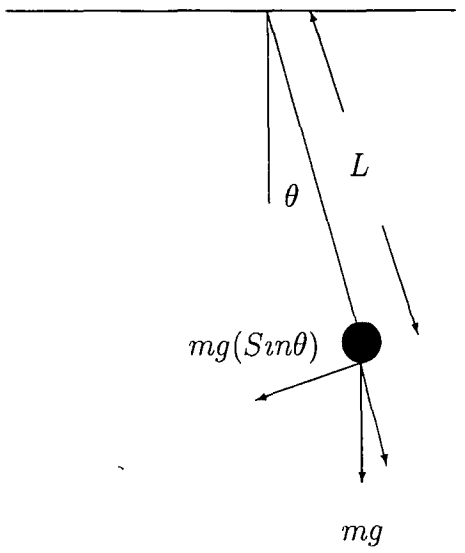


Figure 2 5 Analysis of the Simple Pendulum, an example of a non-linear system

2.8.1 Linearisation

Consider the state space representation of a non-linear system

$$X = F(X,U)$$

$$Y = G(X,U)$$

where $F(\ ), G(\ )$ are non linear functions of the state variables X and input variables U A simple example is the pendulum, as in Fig 2 5 Selection of state variables yields $X_1 = \theta$, which is the pendulum angular displacement and $X_2 = \theta$ which is the pendulum angular velocity Assuming that there is no input signal to the system, analysis from first principles yields

$$\begin{pmatrix} X_1 \\ X_2 \end{pmatrix} = \begin{pmatrix} X_2 \\ -\frac{g}{L}Sin(X_1) \end{pmatrix} = F(X,U)$$

$$Y = X_1 = G(X,U)$$

An equilibrium point can be loosely defined as a position where the system can come to rest In the case of the pendulum this is when

$$X_1 = X_2 = 0$$

A value of state vector X and input vector U where equilibrium can occur will be denoted by asterisks X^* and U^* respectively In this case the point X^* corresponds

to the specific state vector $X_1 = X_2 = 0$. Clearly, for this choice of state vector

$$F(X^*, U^*) = 0$$

and so it is indeed an equilibrium point. Consider system behaviour about small deviations $X^* + x$ from the equilibrium point X^*. It is assumed that the system is differentiable about the equilibrium point so that Taylor's theorem can be applied.

$$F(X^* + x, U^* + u) \approx F(X^*, U^*) + x\left(\frac{\partial F}{\partial X}\right)|_{X=X^*} + u\left(\frac{\partial F}{\partial U}\right)|_{U=U^*}$$

where the approximation symbol indicates that higher order terms have been dropped. A linearised model of this system is thus given by

$$\dot{x} = Ax + Bu$$

$$y = Cx + Du$$

where

$$A = \left(\frac{\partial F}{\partial X}\right)|_{X=X^*} = \begin{bmatrix} 0 & 1 \\ -\frac{g}{L} & 0 \end{bmatrix}$$

is the Jacobian of the function F evaluated at the equilibrium point X^*. Since there is no input and the equation for Y is already linear it can be seen that

$$B = 0$$

$$C = \begin{bmatrix} 1 & 0 \end{bmatrix}$$

$$D = 0$$

2.9 Summary

In this chapter the necessary tools for the computation of the convex estimate μ_{co} of μ have been introduced. The geometric form of the Hahn-Banach Theorem has been shown to provide an appealing framework for the solution of this problem. Two methods for calculating μ_{co} using the 2-norm have been introduced. The limitations of these methods motivate the development of the algorithm outlined in the next chapter. Existing commercial code for μ-analysis has been reviewed. The principal restrictions of this existing code have been highlighted. Finally, L_1 and system linearisation results that are used in the applications chapters of this work have been introduced.

Chapter 3

New Algorithm

The previous chapter outlined a primal and dual method, based on the Multiform numerical range theory, for the calculation of μ_{co} Both methods used the 2-norm This chapter is devoted to an alternative computing strategy for μ_{co} This method also uses the numerical range formulation However a 1-norm dual optimisation strategy is now used, in the belief that it provides the best framework for computing μ_{co} The theory behind this new algorithm is developed and its implementation is described in detail A proof of convergence is given Finally a series of amendments to the basic algorithm are discussed which significantly reduce the time required to obtain tight bounds on μ_{co}

3.1 Dual Problem

It has been shown already that

$$RS \Leftrightarrow 0 \notin W(\alpha^2) \qquad \forall\omega$$

Hence

$$RS \Leftarrow 0 \notin Co(W(\alpha^2)) \; \forall\omega$$

where a larger α pushes the (convex) set $Co(W)$ closer to 0 The explicit reference to α *and* ω will be supressed subsequently The dual method depends on the approximation of the set $Co(W)$ by a polytope P Each iteration of the method produces a

new polytope edge, where at all times

$$P \subset Co(W)$$

Denote the minimum distance (in the 1-norm sense) from 0 to $Co(W)$ as $c_1(\alpha)$ The symbol z will be used to denote a vertex of P Therefore

$$c_1(\alpha) \leq \|z\|_1$$

A primal strategy would attempt to home in on $c_1(\alpha)$ by locating new vertices of P that would minimise $\|z\|_1$ This section looks at the dual of this problem, which makes use of the Hahn-Banach Theorem

3.1.1 Application of Geometric Hahn-Banach Theorem

Consider the case where $c_1(\alpha)$ is some positive number i e ,

$$c_1(\alpha) = \nu > 0 \quad \Leftrightarrow \quad 0 \notin Co(W)$$

This is true

$$\Leftrightarrow Co(W) \cap \nu B_1^\circ = \emptyset$$

where B_1° denotes the interior of the unit ball in the 1-norm sense, i e ,

$$B_1^\circ = \{x \in \mathcal{C}^m \mid \|x\|_1 < 1\}$$

From the Hahn-Banach Theorem there exists a separating hyperplane for these two sets i e ,

$$\exists d \neq 0 \mid < d, w_{co} > \geq < d, \nu b > \qquad \forall w_{co} \in Co(W) \ , \ \forall b \in B_1$$

Noting that, since $\| \|_\infty$ is the dual of $\| \|_1$, this is true

$$\Leftrightarrow < d, w_{co} > \geq \nu \|d\|_\infty \qquad \forall w_{co} \in Co(W)$$

Consider a support hyperplane for P This plane will be denoted by $\hat{d} \in B_\infty$ where $B_\infty = \{x \in \mathcal{C}^m \mid \|x\|_\infty \leq 1\}$ Introduce λ_{min}, the minimum eigenvalue of the hermitian matrix $\sum_{k=1}^m \hat{d}_k A_k$ Therefore, letting ∂B_2 denote the boundary set of the 2-norm unit ball,

$$\lambda_{min}(\sum_{k=1}^m \hat{d}_k A_k) \leq \sum_{k=1}^m \hat{d}_k (x^* A_k x) \qquad \forall \ x \in \partial B_2 \tag{3 1}$$

$x^* A_k x$ is the kth element of a vector in the set W Therefore,

$$\lambda_{min} \leq < \hat{d}, w > \qquad \forall w \in W$$

$$\Rightarrow \lambda_{min} \leq < \hat{d}, z > \qquad \forall z \in P$$

The algorithm will iteratively yield extra vertices for P The beauty of a dual approach lies in the fact that any vertex of P will provide a global lower bound on $c_1(\alpha)$ It is clear that $P \subset Co(W)$ since

$$\lambda_{min} \leq \|\hat{d}\|_\infty \|w_{co}\|_1 \qquad \forall w_{co} \in Co(W) \tag{3 2}$$

This suggests that finding a positive λ_{min} is a valid way of determining whether 0 is in the convex hull of W i e ,

$$\lambda_{min} \geq 0 \quad \Leftrightarrow \quad 0 \notin Co(W)$$

Therefore, this approach yields bounds on either side of $c_1(\alpha)$ at each iteration λ_{min} is a lower bound and $\|z\|_1$ is an upper bound

Remark 1 It is reasonably straight forward to extend the Hahn-Banach Theorem to harden these inequalities into equalities For any positive ϵ

$$c_1(\alpha) \geq \epsilon \Leftrightarrow \exists d \in B_\infty \mid \lambda_{min}(\sum_{k=1}^{m} d_k A_k) \geq \epsilon$$

However for any positive λ_{min} that exists, it can be seen that

$$\max_{d \in B_\infty}(\lambda_{min}(\sum_{k=1}^{m} d_k A_k)) \geq c_1(\alpha) \tag{3 3}$$

Combining expressions (3 2) and (3 3) yields the important result,

$$\max_{d \in B_\infty}(\lambda_{min}(\sum_{k=1}^{m} d_k A_k) = c_1(\alpha) \tag{3 4}$$

Remark 2 The constraint that the hyperplane d be on the boundary of the unit ball (in the infinity norm sense) is in fact no constraint at all as a simple rescaling shows that

$$\max_{d \in B_\infty}(\lambda_{min}(\sum_{k=1}^{m} d_k A_k)) = \max_{d \neq 0}(\lambda_{min}(\sum_{k=1}^{m} \frac{d_k}{\|d\|_\infty} A_k))$$

Comments on the Separating Hyperplane d

Throughout this work $d \in B_\infty$ This means that $d(i) \in [-1, 1]$, $\forall i$ It can be shown [Doyle 82] that for uncertain parameters that are purely complex it is sufficient to determine a strictly positive d i e , $d(i) \in [0, 1]$ This is due to the absence of equality constraints in the complex only case This has speed implications for calculation of complex μ as it leads to a reduction in the number of constraints involved in the optimisation problem under consideration

3.2 Outline of the Algorithm

Motivated by eqn (3 4) a new algorithm for the computation of μ_{co} is now formulated The algorithm will consist of an outer and an inner loop The outer loop selects values of α which are candidate upper or lower bounds on μ_{co} The new α should be selected so as to guarantee convergence of the outer loop and to quickly provide good bounds on μ_{co} The inner loop takes this value of α and answers the basic proximity question (i e , estimates $c_1(\alpha)$) by maximising eqn (3 4) The following subsections briefly outline the steps involved in each loop when calculating complex μ_{co} A more detailed exposition, along with possible bolt-on improvements, is provided in subsequent sections

3.2.1 Outer Loop

The first step in the outer loop is initialisation The outer loop's function is the generation of the next value of α for the inner loop

Initialisation

When initialising, singular vectors of M are used to provide starting constraint z's Singular vectors are attractive for a number of reasons The U, Σ, V matrices that are generated by the Singular Value Decomposition (SVD) are easily obtained Engineers tend to have an intuitive understanding of singular values/vectors as the principal

gains of a system in a MIMO setting They are also quite useful, as they yield bounds on μ_{co}

Generation of $\alpha^{(n+1)}$

Let k_m^{up} and k_m^{low} correspond to upper and lower bounds on $k_{m_{co}}$ respectively After any iteration of the inner loop process one of the following decisions can be taken Either

$$c_1(\alpha) = 0 \;\Rightarrow\; k_m^{up} = \alpha^{(n)}$$

or

$$c_1(\alpha) > 0 \;\Rightarrow\; k_m^{low} = \alpha^{(n)}$$

When $c_1(\alpha) > 0$ another useful lower bound on $k_{m_{co}}$ can be obtained from consideration of the support hyperplane $\hat{d}$ that yields the positive λ_{min} $\hat{D} = \mathbf{diag}(\hat{d})$ is a good scaling matrix when attempting to solve the $\bar{\sigma}(DMD^{-1})$ minimisation problem for μ_{co} Thus,

$$c_1(\alpha) > 0 \;\Rightarrow\; k_m^{low} = \max[(\bar{\sigma}(DMD^{-1}))^{-1}\ ,\ \alpha^{(n)}]$$

The next value of seed α is chosen to be $(k_m^{low} + k_m^{up})/2$ Continuing in this fashion will bisect the interval $[\ k_m^{low}, k_m^{up}\ ]$ at each iteration Use of such a binary search strategy means that convergence of the outer loop is guaranteed Fig 3 1 provides an illustration of the outer loop algorithm

3.2.2 Inner Loop

Manipulation of eqn (3 4) shows that the function to be maximised is equivalent to

$$\max\ \lambda_{min} \ \mathbf{subject\ to}\ \ < d, z > \geq\ \lambda_{min}$$

The above expression is clearly a linear program Each iteration of the linear program will provide a plane d which acts as a separating hyperplane between 0 and the polytope P If no such hyperplane can be found then $0 \in Co(W)$ The next task is to find a new z or z's that will maximise λ_{min} An eigenvalue/eigenvector decomposition, also known as the Characteristic Value Decomposition **(CVD)** will provide new z

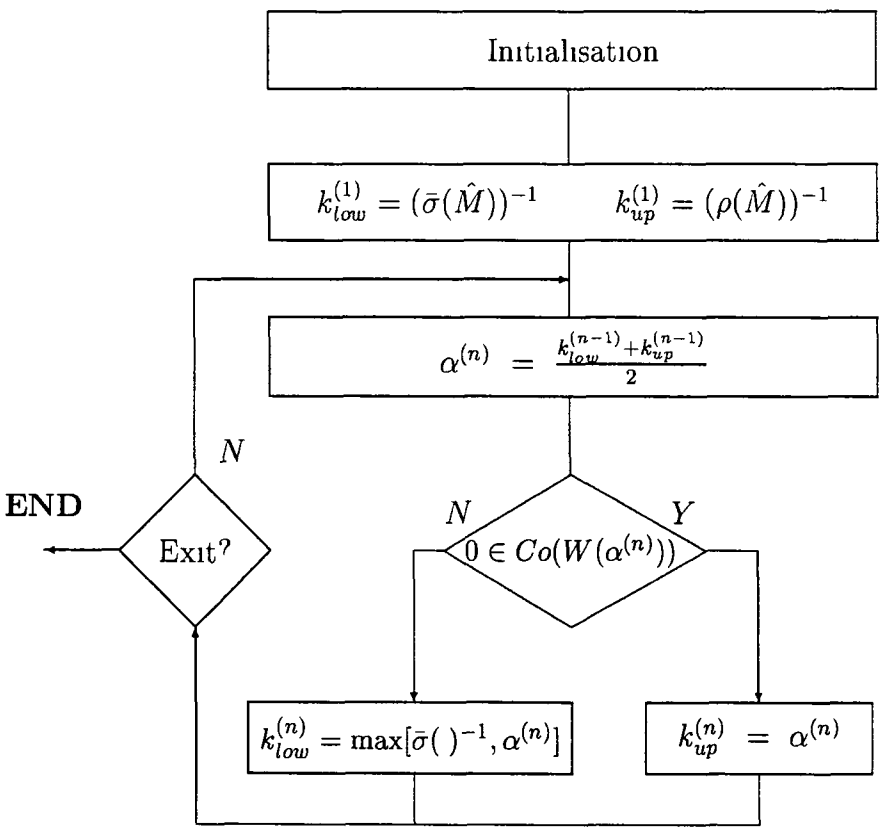


Figure 3 1 Flow Diagram for the Outer Loop of the New Algorithm

vectors that, when added to the constraint list P for the next iteration of the linear program, will increase λ_{min} There will be three ways of exiting the inner loop in the basic algorithm

1 $\lambda_{min} > 0$ When the minimum eigenvalue of the $\sum_{k=1}^{m} d_k A_k$ matrix is positive it implies that $c_1(\alpha) > 0$ Thus, it can be concluded that $\frac{1}{\alpha} > \mu_{co}$

2 **LP returns** $d = 0$ When the Linear Program cannot find a non-zero support hyperplane for the current list of z's, this implies that $c_1(\alpha) = 0$ Thus, $\frac{1}{\alpha} \leq \mu_{co}$

3 **Maximum iteration count exceeded** This optional third exit criterion can be included if rough bounds are required quickly It precludes the possibility of having too many iterations of the linear program when the value of α is very close to μ_{co}

Fig 3 2 provides an illustration of the basic inner loop algorithm

3.3 Inner Loop Analysis and Some Suggested Improvements

This section details the necessary work involved in the various inner loop steps It concludes with some amendments to the basic algorithm that result in a significant improvement in computing times A quantitative analysis of how these various amendments improve the basic algorithm is presented in the next chapter

3.3.1 The Linear Program (LP)

The function to be maximised, $\lambda_{min}(\sum_{k=1}^{m} d_k A_k)$, can be formally stated as the LP

$$\max_{d \in B_\infty} \lambda \quad \textbf{subject to}$$

1 $\|d\|_\infty \leq 1$ and

2 $< d, w_{co} > \geq \lambda \qquad \forall w_{co} \in Co(W)$

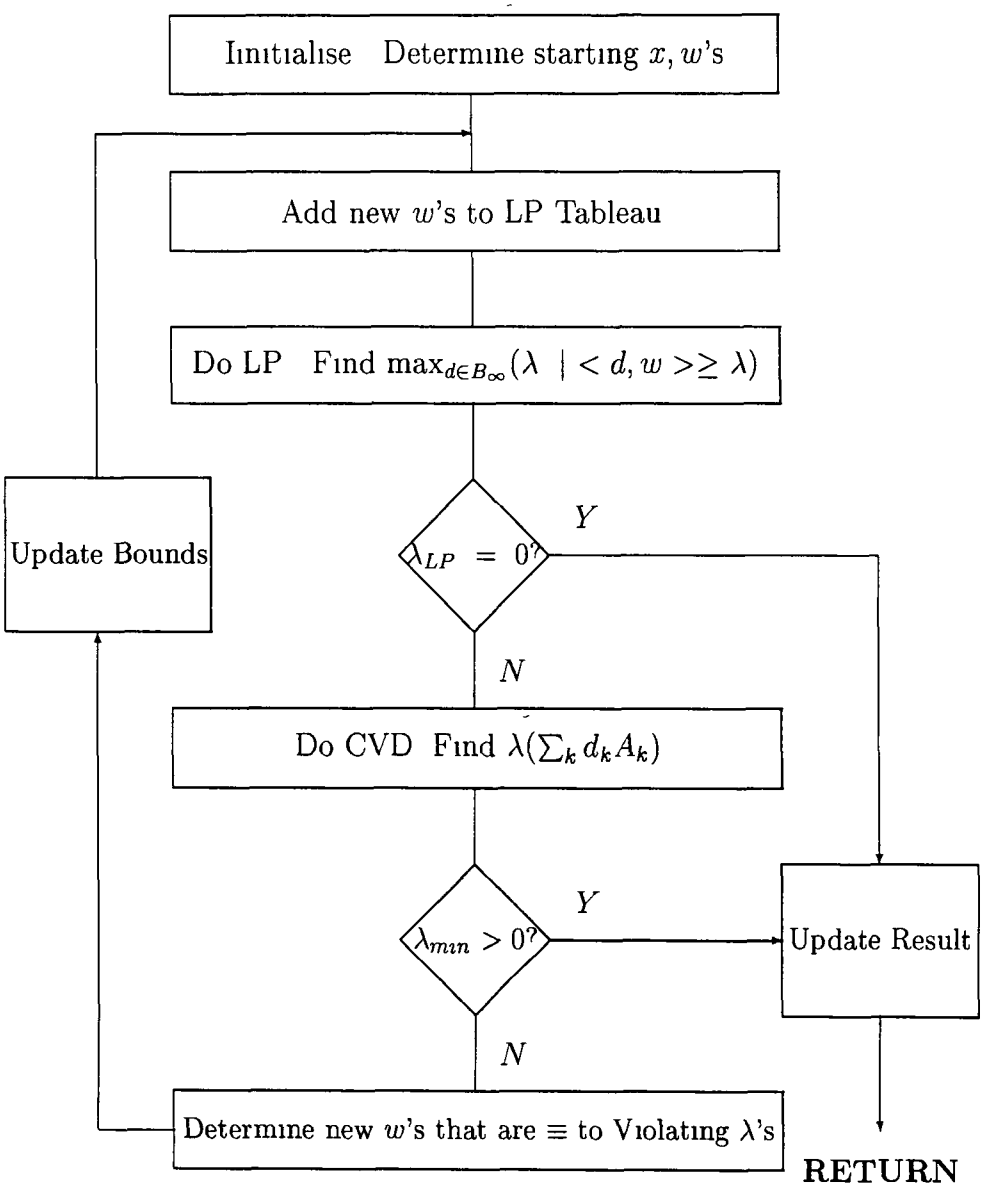


Figure 3 2 Flow Diagram for the Inner Loop of the New Algorithm

Note that constraint 1 can be recast as

$$-1 \leq d_k \leq +1 \qquad k = 1, \quad , m$$

or equivalently,

$$d_k \leq +1 \quad \textbf{and} \quad -d_k \leq +1 \qquad k = 1, \quad , m$$

In a similar fashion constraint 2 can be restated as

$$\lambda - \ < d, w_{co} > \leq \ 0 \qquad \forall w_{co} \in Co(W)$$

Each of these two sets of constraints are linear and therefore this can be viewed as a linear program in $m+1$ variables where the first variable λ is the only one of interest to the objective function Of course viewing each vector in $Co(W)$ as a constraint is problematic as there are an infinite number of them The approximation of $Co(W)$ by the finite polytope P will *relax* the requirement on λ, because an infinite number of constraints is turned into a finite list of (hopefully good) z's Since $P \subset Co(W)$, the λ that will be maximised by such a constraint list will provide an upper bound on $c_1(\alpha)$ This solution λ from the LP will be denoted from now on by λ_{LP} Assume that there are N of these constraints The linear program to be solved is then

$$\max \ C^T \mathbf{x} \quad \textbf{subject to}$$

$$A \mathbf{x} \quad \leq 0$$

where

$$x^T \ = \ (\lambda, d_1, \quad , d_m)$$

$$C^T \ = \ (1, 0_1, \quad , 0_m)$$

$$A = \{A_1, A_2, A_3\}$$

There are three blocks which are stacked on top of each other to construct the constraint tableau A The first block contains the finite list of vertices that have been accumulated thus far

$$A_1 \ = \ \begin{pmatrix} 1 & -(z^{(1)})^T \\ & \\ 1 & -(z^{(N)})^T \end{pmatrix}$$

The second two blocks are concerned with constraining the solution plane d to be inside the unit ball in the infinity norm sense

$$A_2 = \begin{pmatrix} 0 & 1 & 0 & , & , & 0 \\ 0 & 0 & 1 & , & , & 0 \\ & & & & & \\ 0 & 0 & 0 & , & , & 1 \end{pmatrix} \qquad A_3 = \begin{pmatrix} 0 & -1 & 0 & , & , & 0 \\ 0 & 0 & -1 & , & , & 0 \\ & & & & & \\ 0 & 0 & 0 & , & , & -1 \end{pmatrix}$$

The standard linear program solver used throughout this work is an amended version of the public domain Numerical Recipes LP code offered in Flannery *et al* [Flannery 91] This version was implemented in *MATLAB* The code was found to offer a good tradeoff between reliable operation and good speed It incorporates an anticycling algorithm and was significantly better than the linear program solver offered in *MATLAB*'s optimisation toolbox The Numerical Recipes code attempts to minimise rather than maximise a given objective function This is not a restriction as clearly,

$$\max_d \; C^T \mathbf{x} \quad \equiv \quad \min_d \; -C^T \mathbf{x}$$

3.3.2 CVD

It is natural to consider the eigenvector/eigenvalue decomposition of the matrix $\sum_{k=1}^{m} d_k A_k$ since

$$c_1(\alpha) \; = \; \max_d \; \lambda_{min} \left(\sum_{k=1}^{m} d_k A_k \right)$$

This means that

$$\sum_{k=1}^{m} d_k A_k \; \geq \; \lambda_{min} I$$

$$\Rightarrow \sum_{k=1}^{m} d_k x^* A_k x \; \geq \; \lambda_{min} \qquad \forall x \in B_2$$

where the unit eigenvector(s) $\hat{x}$ corresponding to λ_{min} achieve equality Thus

$$\sum_{k=1}^{m} d_k w_k \; \geq \; \lambda_{min} \qquad \forall w \in W$$

$$\Rightarrow \sum_{k=1}^{m} d_k w_{co_k} \; \geq \; \lambda_{min} \qquad \forall w_{co} \in Co(W)$$

$$\Rightarrow \sum_{k=1}^{m} d_k z_k \geq \lambda_{min} \qquad \forall z \in P$$

Therefore this algorithm provides a global lower bound on $c_1(\alpha)$ The CVD is useful here because it yields the candidate unit eigenvector $\hat{x}$ whose corresponding eigenvalue is a lower bound on λ_{min} It is very convenient to perform the CVD because *MATLAB* is designed to provide numerically reliable eigenvalue/eigenvector decomposition code for hermitian matrices

3.3.3 Amendments to the Inner Loop

This section looks at ways in which the inner loop proximity question can be solved more quickly The amendments mentioned here are equally applicable to real and complex uncertainties

Constraint Generation

The biggest single improvement that can be made to the basic algorithm is to ensure that "good" unit vectors do not need to be regenerated during subsequent iterations of an inner loop The motivation for this work can be seen in Fan *et al* [Fan 2 86] with respect to the geometric aspects of the numerical range formulation It can be seen that different values of seed α act, loosely speaking, as a translational operator on the set W Therefore, unit vectors which produce w's of small norm tend to carry over for different seed α's Storage and reuse of such good unit vectors means that the number of iterations of the inner loop that are required is cut dramatically

Constraint Pruning/Plateaus

Consider four successive stages of the search for a positive minimum eigenvalue in Fig 3 3 Notice the series of plateaus that λ_{min} goes through during the course of one seed α While λ_{min} is stuck on this plateau, improvement is occurring in the upper bound on $c_1(\alpha)$ New "tight" unit vectors are being found which are generating w vectors of smaller norm The dual algorithm seems to exhaust the benefit of a new eigenvalue for a few LP/CVD loops before moving onto a new level Once on a new

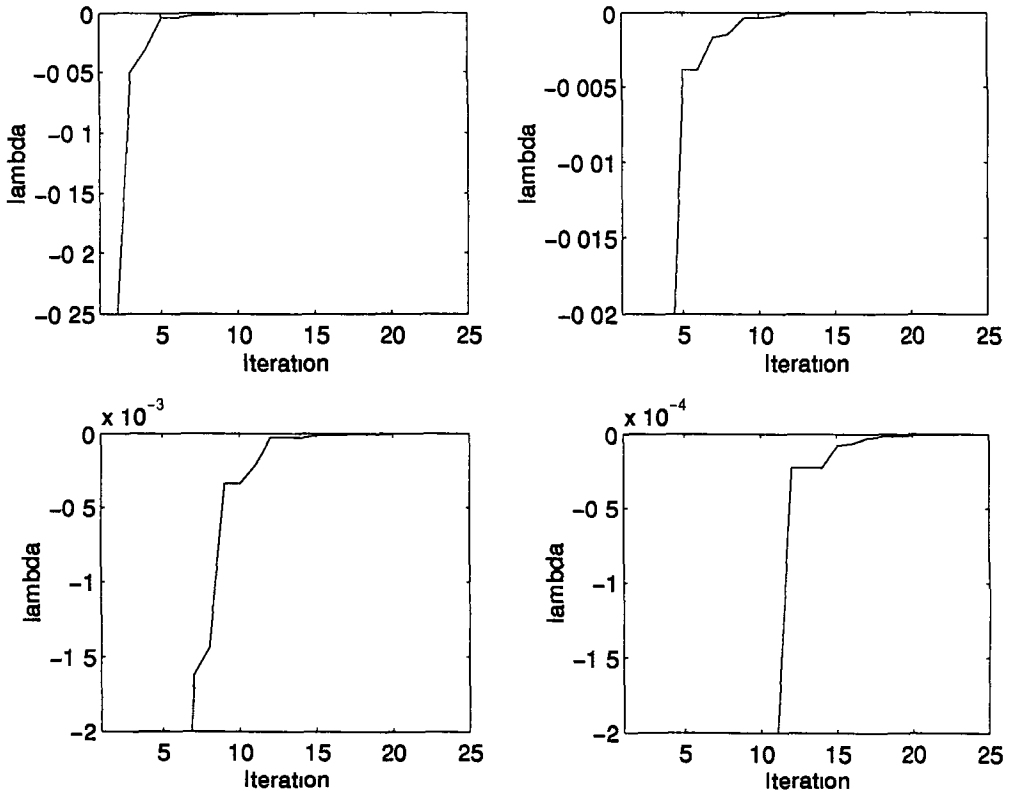


Figure 3 3 Progress of λ_{min} for a value of α that yields 0 close to the boundary of $Co(W)$

plateau, the unit vectors that were tight on the previous plateau now begin to lose their effectiveness

When an outer loop iteration has been completed there are two possible approaches that can be taken which will reduce the size of the LP tableau The first approach is to discard unit vectors that are generating w's that are a considerable distance (rule of thumb $\geq 20\%$) away from the unit vector that has generated the "best" w A second possibility is to say that only the unit vectors which cause plateau jumps in λ_{min} are of interest This approach can work in tandem with the previous approach i e , one can still get rid of unit vectors that are not producing good w's While the former approach proved to be quite successful and was integrated into the improved form of the algorithm, computational experience has shown that use of the latter approach does not effect any significant reduction in computing times

Tableau Formation

The Numerical Recipes LP solver includes the facility to deal with degeneracy Notwithstanding this, the presence of lots of zeros in the problem tableau is not a good idea as they tend to introduce numerical problems This is particularly likely to occur when dealing with real/mixed uncertainty problems One possible improvement is to rearrange the linear programming problem so that the solution plane produces inner products that are around unity rather than zero Up to now the search has been for the minimum eigenvalue that acts as a lower bound in the following way

$$c_1(\alpha) \geq \lambda_{min} = \min_{x \in \partial B_2} x^*(\sum_k d_k A_k)x \qquad d_k \in [-1, +1] \tag{3 5}$$

Now consider the new plane $f_k = d_k + 1$ An exactly similar problem to (3 5) is

$$\lambda_{min} \leq \min_{x \in \partial B_2} x^*(\sum_k f_k A_k - A_k)x \qquad f_k \in [0, 2] \tag{3 6}$$

$$\Leftrightarrow \lambda_{min} \leq \sum_k (f_k w_k - w_k)$$

$$\Leftrightarrow \lambda_{min} - < f, w > \leq -\sum_k w_k$$

This linear program is intuitively more appealing Using variables which are strictly positive ties in better with classical linear programming ideas There are also fewer

zeros in the LP tableau The extra software overhead, namely calculating and storing $\sum_k w_k$, is also not expensive

3.4 Improvements to the Basic Algorithm

Numerical experience has shown that the speed of the algorithm is greatly improved by certain additions, which are the subject of this section Firstly, a prescaling routine that is performed on the problem matrix during initialisation is considered Secondly, amendments to the outer loop are discussed These amendments preserve convergence to μ_{co}, but at a rate faster than that given by a binary search The following two possibilities are considered -

(i) Information that can be gained from the list of w_{co}'s that are "close" to being orthogonal to the support plane d

(ii) New bounds that can be deduced from matrix pencil theory

For the present, the uncertainty is taken to be complex Comments about improvements which pertain directly to the real/mixed uncertainty problem are made at the end of the section A quantitative analysis of these improvements is presented in Chapter 4

3.4.1 Initialisation

When initialising the algorithm, the unit vectors which generate the z's are of particular concern It is desirable to quickly find "good" unit vectors that correspond to z's which are close to, or are on the boundary of $Co(W)$ However there is no reason to expect that the SVD singular vectors of M will generate constraint z's that are close to the boundary of $Co(W)$ Computational experience has shown that the best way to do this initially is to pre-condition the matrix M in question using a scaling algorithm due to Osborne [Osborne 60] This process produces a new matrix $\hat{M}$ which has the following properties

1 $\mu_{co}(\hat{M}) = \mu_{co}(M)$

2 The condition number, $\kappa(\hat{M})$, of the matrix $\hat{M}$ is minimised

3 $\mu_{co}(M) \leq \bar{\sigma}(\hat{M}) \leq \bar{\sigma}(M)$

In practice, using the matrix $\hat{M}$ produced by Osborne's scaling routine provides (i) a much tighter initial upper bound on $\mu_{co}(M)$ and (ii) a much better set of starting vectors for the inner loop To see this consider the basic version of the preconditioning method due to Osborne [Osborne 60] which is outlined in Fig 3 4

Nomenclature

R_j is the 2-norm of the jth row vector *without* the $M(j,j)$ element
S_j is the 2-norm of the jth column vector *without* the $M(j,j)$ element
$\bar{D} = \mathbf{diag}(1, 1, \quad, (\frac{S_j}{R_j})^{-\frac{1}{2}}, 1, \quad, 1)$

This method attempts to "diagonalise" the matrix M using similarity transformations by iteratively reducing the size (norm) of the off diagonal elements The basis for this approach lies in results from eigenvalue sensitivity theory [Golub 89] Minimising the condition number of the scaled matrix DMD^{-1} is desirable since

$$\underline{\sigma}(DMD^{-1}) \leq \mu_{co} \leq \bar{\sigma}(DMD^{-1})$$

Hence, reducing $\kappa(DMD^{-1})$ will tighten the bounds on μ_{co}

3.4.2 Matrix Pencils

An application of the Generalised Characteristic Value Decomposition **(GCVD)** [Golub 89], [Gantmacher 60] to the problem at hand is now described Consider a positive quadratic form i e ,

$$\sum_{k=1}^{m} d_k A_k > 0$$

Each A_k can be viewed as the sum of two matrices, one that is independent (B_k) and one that is dependent (C_k) on α Thus,

$$A_k = B_k - \alpha^2 C_k$$

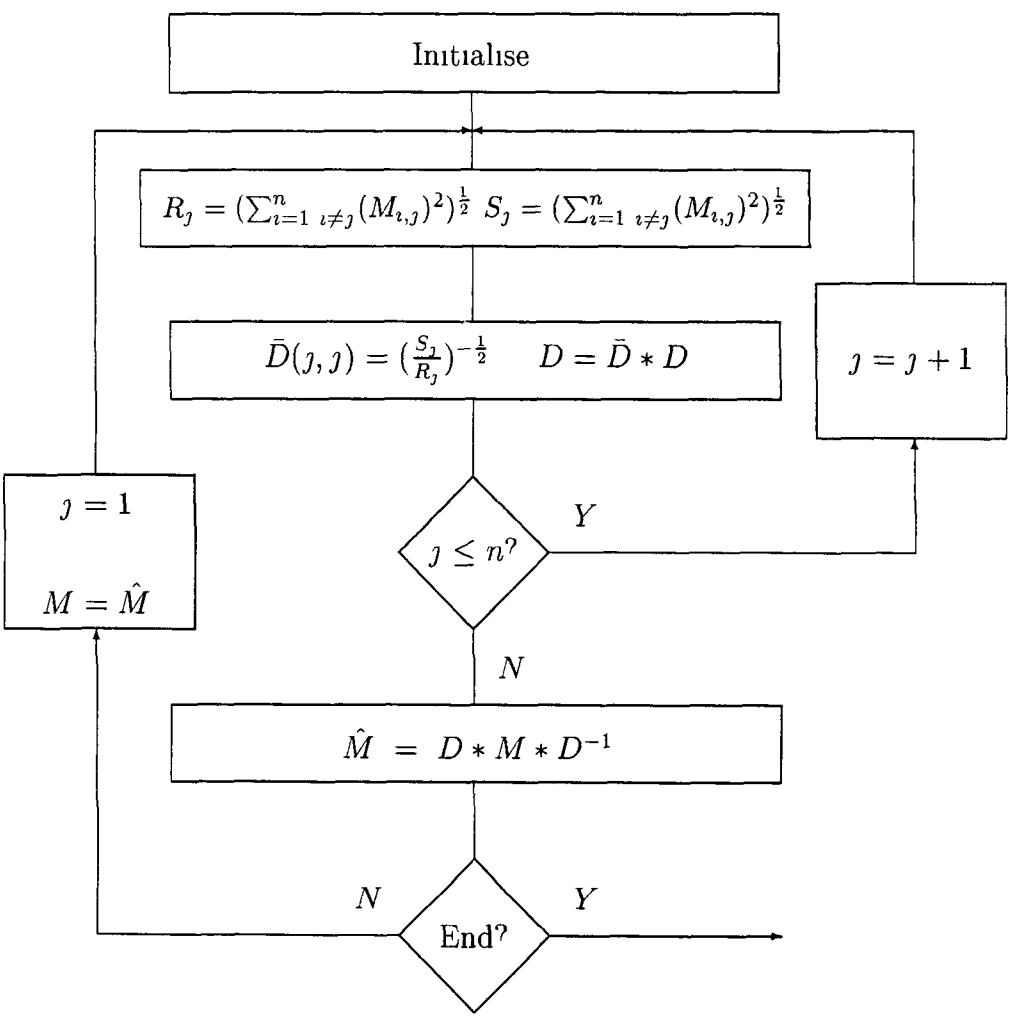


Figure 3 4 Flow diagram for Osborne's preconditioning algorithm

An equivalent representation of the form is then

$$\sum_{k=1}^{m} d_k(B_k - \alpha^2 C_k) > 0 \tag{3 7}$$

Consider the C_k block that is dependent on α Increasing the value of α sufficiently will produce an indefinite form when $\sum_{k=1}^{m} d_k C_k$ is of full rank, i e ,

$$\exists \delta \mid \lambda_{min}\left(\sum_{k=1}^{m} d_k(B_k - (\alpha^2+\delta)C_k)\right) = 0$$

One can think of $\delta(\sum_{k=1}^{m} d_k C_k)$ as being a clearance factor Thus

$$\lambda_{min}\left(\sum_{k=1}^{m} d_k(B_k - \alpha^2 C_k) - \delta(\sum_{k=1}^{m} d_k C_k)\right) = 0 \tag{3 8}$$

Consider this matrix pencil First note that there is a unitary matrix U such that

$$\sum_{k=1}^{m} d_k(B_k - \alpha^2 C_k) = U\Lambda U^*$$

where Λ is a diagonal matrix Thus eqn (3 8) is equivalent to asking whether the matrix

$$U\Lambda U^* - \delta\mathcal{C}$$

is singular, where $\mathcal{C} = \sum_{k=1}^{m} d_k C_k$ Pre and post multiplication yields

$$\Lambda^{-\frac{1}{2}} U^*(U\Lambda U^* - \delta\mathcal{C})U\Lambda^{-\frac{1}{2}} \tag{3 9}$$

Note that this operation does not preserve the eigenvalues of the original matrix However, it does not affect their sign or its rank Expression (3 9) has the following equivalent representations,

$$\mathcal{I} - \delta\Lambda^{-\frac{1}{2}} U^* C U \Lambda^{-\frac{1}{2}}$$

taking the SVD again yields

$$= \mathcal{I} - \delta V \Lambda_C V^*$$

where Λ_C is a diagonal matrix Pre and post multiplying by V^* and V gives

$$= V^*(\mathcal{I} - \delta V\Lambda_C V^*)V$$

$$= \mathcal{I} - \delta\Lambda_C$$

The eigenvalues of Λ_C can be read directly The smallest δ that will singularise the original matrix will therefore correspond to

$$\delta_{MP} = \min_{\lambda \in \Lambda_C}(\lambda^{-1})$$

To avail of this result remember that determination of any positive λ_{min} yields a lower bound on the maximum possible minimum eigenvalue

$$\lambda_{min}^{(n)} \leq \lambda_{min}^{(n+1)} \leq \max_{d}(\lambda_{min}) = c_1(\alpha)$$

Time considerations dictate that the inner loop always terminates before attaining this maximum Up to this point knowing whether λ_{min} is positive has been sufficient By letting the inner loop run, the gap between λ_{min} and λ_{LP} gets smaller The above matrix pencil theory shows how a tight, positive lower bound on $c_1(\alpha)$ can be viewed as a bound on how far a form can be translated from its current position whilst still preserving its positive character This means that a new improved lower bound on $k_{m_{co}}$ is

$$k_{m_{low}}^{(n)} = ((\alpha^{(n)})^2 + \delta_{MP})^{\frac{1}{2}}$$

3.4.3 Tight Constraints

Further improvements to the upper and lower bounds on μ_{co} can be made by considering additional information that can be deduced from good constraints It should be noted that the bounds derived in this subsection 3 4 3 are not rigorous, i e , it is impossible to say whether the new bound is an upper or lower one on μ_{co} Despite this drawback it is possible to gain information that will improve on the speed of a binary search

Non-zero λ_{LP}

Consider the finite constraint list that is generated during the inner loop Furthermore, consider the inner product generated by the solution hyperplane d with an arbitrary vertex $z^{(i)}$

$$< d, z^{(i)} > \geq \lambda_{LP} \geq c_1(\alpha)$$

The current value of λ_{LP} can be used as a scaling parameter that will translate the current polytope to a position where 0 is a vertex Therefore the current plane d no longer separates 0 from $Co(W)$ Considering expr (3 7) it can be seen that

$$< d, z_B^{(i)} - \alpha^2 z_C^{(i)} > \geq \lambda_{LP}$$

$$\Rightarrow \exists \delta \mid < d, z_B^{(i)} - (\alpha^2 + \delta) z_C^{(i)} > = 0$$

In effect some vertex of the polytope is being translated to 0 The aim is to find the smallest δ required which, when acting as an offset for the current α, would enable 0 to be a boundary point of $Co(W)$ i e , $0 \in \partial(Co(W))$ Note that this does NOT mean that there may not exist some other plane d which may act as a suitable separating hyperplane For this reason it cannot be said whether this offset provides an upper or lower bound on μ_{co} However, depending on how close the gap is between λ_{LP} and λ_{min} it can be expected that this new bound on μ_{co} will be quite tight Another iteration of the inner loop is necessary to determine what kind of bound it is Thus, what is required is the smallest δ so that

$$\delta < d, z_C^{(i)} > \geq \lambda_{LP} \qquad \forall i$$

$$\Leftrightarrow \delta \geq \frac{\lambda_{LP}}{< d, z_C^{(i)} >} \qquad \forall i$$

Therefore

$$\alpha^{(n+1)} = ((\alpha^{(n)})^2 + \delta)^{\frac{1}{2}}$$

where δ is generated according to the rule

$$\min \delta \mid \delta \geq \frac{\lambda_{LP}}{< d, z_C^{(i)} >} \qquad \forall i$$

Note that knowledge of the z vectors where $< d, z_C^{(i)} >$ is close to δ provides a suitable basis for pruning constraints This keeps the amount of good constraints that are brought from iteration to iteration (and thus tableau size) manageable

Information From λ_{min}

Upper bound information can also be gained from an accurate value for λ_{min} once it is known whether $c_1(\alpha)$ is zero or not Remember that from expr (3 7)

$$\sum_{k=1}^{m} d_k (B_k - \alpha^2 C_k) \geq \lambda_{min} I$$

$$\Rightarrow < d, z_B^{(i)} - \alpha^2 z_C^{(i)} > \geq \lambda_{min} \qquad \forall i$$

Case 1: $c_1(\alpha) > 0$ **thus** λ_{min} **Positive**

The task is to increase α^2 by δ so as $0 \in \partial Co(W)$ for this choice of plane d This problem is equivalent to finding the smallest δ that will reduce λ_{min} to zero for the finite list of tight constraints that have been accumulated Using a similar argument to before

$$< d, z_B^{(i)} - (\alpha^2 + \delta) z_C^{(i)} > \geq 0$$

$$\Leftrightarrow \delta < d, z_C^{(i)} > \leq \lambda_{min} \qquad \forall i$$

This means that the smallest δ required to push λ_{min} negative for the current d is given by

$$\delta = \max_i \frac{\lambda_{min}}{< d, z_C^{(i)} >} = \frac{\lambda_{min}}{\min_i < d, z_C^{(i)} >}$$

It should be noted at this point that the list of constraints being assessed to determine δ will be necessarily incomplete This suggests that the smallest δ would be larger than required for the current d in order to push 0 inside $Co(W)$ Again, it is not possible to say whether a plane d exists that will separate 0 from $Co(W)$ Therefore this is another estimate for μ_{co} where a further inner loop iteration is required to determine whether it is an upper or lower bound However, use of good constraints should mean that the new bound

$$\alpha_{(n+1)} = ((\alpha^{(n)})^2 + \delta)^{\frac{1}{2}}$$

can at least improve on a binary search

Case 2: $c_1(\alpha) = 0$ **thus** λ_{min} **Negative**

Here the task is to reduce α^2 by δ so that $0 \in \partial Co(W)$ For the current list of constraints no hyperplane can be found that will separate 0 from $P \subset Co(W)$ Denote the largest (negative) λ_{min} that has been found so far as $\lambda_{min}^{(n)}$ This value will be less than the smallest inner product $< d^*, w^* >$ generated by an optimal hyperplane d^* and the best w^* Consider the non-zero hyperplane $d^{(n-1)}$ that the LP returns on its

penultimate iteration For the finite list of z's that have been computed

$$\lambda_{min}^{(n)} \leq \min_{i} < d^{(n-1)}, z^{(i)} >$$

Now calculating a δ which will make $\lambda_{min}^{(n)}$ positive is only valid for the current list of constraints Therefore finding a δ so that

$$\lambda_{min} + \delta < d, z_C^{(i)} > \geq 0$$

means that a separating hyperplane can be found between 0 and the current polytope P However, it may be possible that another constraint might invalidate this claim Therefore the smallest δ given by

$$\delta = \max_{i} \frac{\lambda_{min}}{< d, z_C^{(i)} >} = \frac{\lambda_{min}}{\min_i < d, z_C^{(i)} >}$$

should provide a tight bound on μ_{co} Again there is no information about whether it is an upper or lower bound Another drawback when $\lambda_{min}^{(n)}$ is negative is the gap that may exist between $\widehat{\lambda}_{min}$ and $\lambda_{min}^{(n)}$ when the inner loop terminates When $\lambda_{min}^{(n)}$ is positive the algorithm can proceed until $\lambda_{min}^{(n)}$ and λ_{LP} are arbitrarily close to each other Since λ_{LP} always acts as an upper bound on $\widehat{\lambda}_{min}$, the $[\widehat{\lambda}_{min}$, $\lambda_{min}^{(n)}]$ gap can be made arbitrarily small This is not true when $\lambda_{min}^{(n)}$ is negative The algorithm terminates when it has determined that $0 \in Co(W)$ At this point the gap between $\widehat{\lambda}_{min}$ and $\lambda_{min}^{(n)}$ may be significant This will have an adverse effect on the quality of the bound on μ_{co} that is obtained

Comparison of Matrix Pencil and Tight Constraint Bounds

Numerical experience has shown that in most cases the matrix pencil theory lower bound provides a better estimate for μ_{co} However, it should be noted that the necessary constraint pruning software provides these bounds as a byproduct Also, by considering only good constraints through the use of a proper pruning strategy it does not take long to ascertain whether these new bounds are upper or lower ones

3.5 Improvements with Real Uncertainty

Improvements to the basic algorithm that are applicable to real/mixed uncertainty problems are now considered

It should be noted that in particular, that the above matrix pencil theory only applies to the case of purely complex uncertainty Also, the information that is derived from tight constraints can be limited by the peculiar geometry of the $Co(W)$ generated in a mixed uncertainty environment Therefore it is necessary to determine improvements on a basic binary search strategy that are of use in the mixed/real case In this subsection a rigorous upper bound on μ_{co} is considered which is applicable to both the purely complex and the mixed case At the end of the section two special cases are briefly considered Speed improvements are possible using sample perturbation bounds and early termination of the linear program The former can be considered an outer loop method whilst the latter should be viewed as an improvement to the inner iteration

Rigorous Lower Bound

The basis for the bound involves consideration of expr (3 7), i e ,

$$\sum_{k=1}^{m} d_k(B_k - \alpha^2 C_k) \geq \lambda_{min}\mathcal{I}$$

The problem at hand is to consider what δ can be added to α^2 so as that λ_{min} will still be positive for the current plane d In this way $(\alpha^2 + \delta)^{\frac{1}{2}}$ will still act as a lower bound on $k_{m_{co}}$ It is necessary to find a δ so that

$$\sum_{k=1}^{m} d_k(B_k - (\alpha^2 + \delta)C_k) \geq 0$$

This will be certainly true if

$$\delta \sum_{k=1}^{m}(d_k C_k) \leq \lambda_{min}\mathcal{I}$$

which in turn will be the case if δ is selected so that

$$\delta = \frac{\lambda_{min}}{\bar{\sigma}(\sum_{k=1}^{m} d_k C_k)}$$

The maximum singular value of the $\sum_{k=1}^{m}(d_k C_k)$ matrix corresponds to its maximum principal gain. Thus, a new lower bound that will allow improvement on a binary search follows if $k_{m_{low}}^{(n+1)}$ is selected according to

$$k_{m_{low}}^{(n+1)} = ((\alpha^{(n)})^2 + \delta)^{\frac{1}{2}} \tag{3.10}$$

Note that no rigorous improvement to a standard binary search has been found when $\alpha^{(n)}$ corresponds to an upper bound on $k_{m_{co}}$

Sample Perturbation Bounds

This subsection considers further refinement of the lower bound on mixed/real μ_{co}. At the moment the standard practice is to find permissible matrices that make the loop $(I + M\Delta)$ singular [Young 95]. In [Young 90] a power method is presented which finds candidate destabilising Δ's. A simpler method which can provide reasonable bounds during initialisation is to assume that at least one complex block exists. In any robust performance problem there will always exist at least one complex parameter. If one assumes that all the permissible real uncertainties assume their nominal values then

$$\mu_{co}(M) \geq \max_i \{M_{ii}\} \qquad \texttt{i complex}$$

where the candidate Δ corresponds to $\mathbf{diag}(0, 0, \ldots, \Delta_i, \ldots, 0, 0)$. A suitable grid search of other perturbation Δ's may also be useful.

Early Linear Program Termination

Determination of a very accurate value of $c_1(\alpha)$ can be a very time consuming exercise from which little benefit may be derived, given the geometry of the mixed uncertainty case. If a positive λ_{min} exists for a certain seed value of α then it could be argued that there may be no benefit in using further computations to find the best plane that will maximise λ_{LP}. Consequently there may be no need to compute the optimal λ_{LP} and λ_{min} for a given batch of constraints. The implementation of this strategy is quite straightforward. The LP solver is amended so that it terminates when any feasible non-zero vector, not necessarily the optimal solution, has been located.

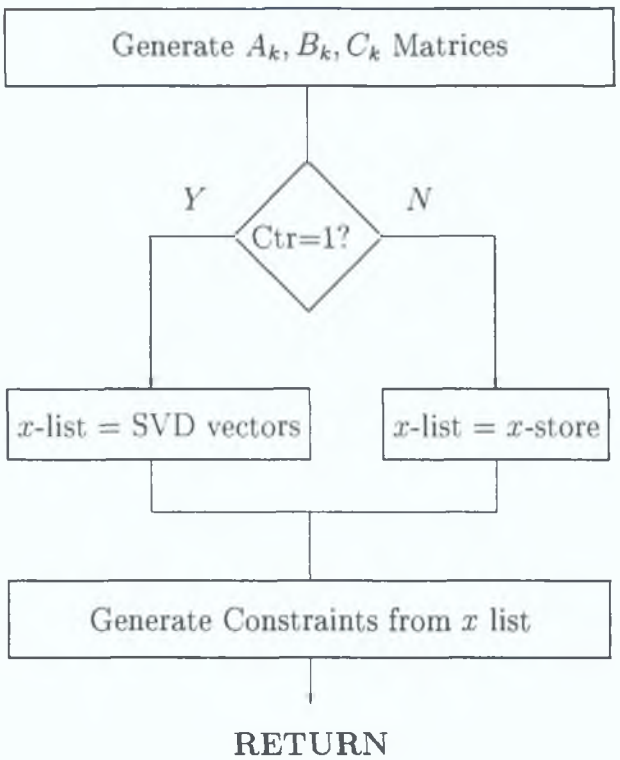


Figure 3.5: Flow Diagram for the Initialisation phase of the Improved Algorithm

The drawback of such a strategy is that the separating d plane does not provide positive λ_{min}'s as quickly during the CVD part of the algorithm. In practice, terminating the LP early has provided very mixed results on a selection of problems. Hence, it is not considered further.

3.6 Implementation of Changes to the Basic Algorithm

This section outlines the algorithmic changes that have been made. Conveniently, the majority of the changes can be seen as simple modifications to the original code. Fig. 3.5 illustrates the new initialisation phase. The inner and outer loop improvements require the generation of B_k and C_k blocks where, as before, $A_k = B_k - \alpha^2 C_k$. In any given iteration of the inner loop the initial constraint tableau is formulated from either existing tight unit vectors from a previous iteration or, when starting the process, from Osborne singular vectors.

It has been found that maximum benefit can be derived from constraint storage and pruning if the following strategy is adopted. When a positive λ_{min} has been located the inner loop continues until a tight bound on $c_1(\alpha)$ has been located. At this point pruning is performed. This produces a small list of tight constraints which provide polytope vertices that are near the boundary of $Co(W)$ in terms of this level of accuracy. Now, where it is applicable, the matrix pencil, tight constraint and singular value bounds outlined in the previous sections are determined. With these good constraints a simpler form of the inner loop is executed where the only question of interest is whether or not $0 \in Co(W)$. This simpler inner loop can be run in conjunction with a straightforward binary search until it is noticed that a significant number of iterations of the inner loop are required before termination. At this point the value of these constraints has been exhausted. To move onto a new plateau it is necessary to once more run the longer form of the inner loop.

This strategy is conveniently implemented by adding new code either side of the basic algorithm. Flow diagrams for this new code are presented in Figs. 3.6 and 3.7.

3.7 Proof of Convergence

The purpose of this section is to prove that the inner loop converges to $c_1(\alpha)$. The outer loop, at worst, can always fall back on a binary search so its convergence is always guaranteed. A convenient reference for the standard real analysis results referred to in this section is [Smith 90]. To prove that the algorithm is convergent, let $\widehat{\lambda}_{min}$ be the solution of the following problem

$$\widehat{\lambda}_{min} = \max_{d \in B_\infty} \lambda_{min}(\sum_{k=1}^{m} d_k A_k) \tag{3.11}$$

and similarly let $\widehat{\lambda}_{LP}$ be the solution of

$$\widehat{\lambda}_{LP} = \max_{d \in B_\infty} \lambda_{LP} \mid \lambda_{LP} \leq < d, w_{co} > \qquad \forall w_{co} \in Co(W) \tag{3.12}$$

These two problems provide lower and upper bounds on $c_1(\alpha)$ respectively. The problem at hand then is to show that

$$\widehat{\lambda}_{min} = c_1(\alpha) = \widehat{\lambda}_{LP}$$

After the nth iteration

$$\lambda_{min}^{(n)} \leq c_1(\alpha) \leq \lambda_{LP}^{(n)}$$

Consider a sequence $\{\lambda_{min}^{(n)}\}_{n=1}^{\infty}$ which is bounded above by $\widehat{\lambda}_{min}$. Let

$$\lambda_{min}^{(n)} = \max\left[\lambda_{min}^{(n-1)}, \lambda_{min}(\sum_{k=1}^{m} d_k^{(n)} A_k)\right]$$

so that the nth minimum eigenvalue in the sequence corresponds to the best that has been found so far. In this way the sequence is bounded, monotone and non-decreasing. Similarly the sequence $\{\lambda_{LP}^{(n)}\}_{n=1}^{\infty}$ is bounded, monotone and non-increasing. To see this, consider that after n iterations of the linear program there are a finite number of constraints such that

$$\lambda_{LP}^{(n)} \leq < d, z^{(i)} > \qquad \forall i$$

Adding a new constraint can only *tighten* the requirement on λ_{LP}. Certainly, since the old constraints are still there the minimum cannot be any higher than it was for a previous iteration. Hence,

$$\lambda_{LP}^{(n+1)} \leq \lambda_{LP}^{(n)}$$

Consider the following lemma for the sequence $\{\lambda_{min}^{(n)}\}$. The argument carries over exactly for $\{\lambda_{LP}^{(n)}\}$

Lemma 3.6.1 *The bounded monotone sequence $\{\lambda_{min}^{(n)}\}$, converges to the maximum described in eqn. (3.11).*

Proof The Bolzano-Weierstrass theorem dictates that such a sequence must have at least one accumulation point. Let such an accumulation point be denoted by $\widetilde{\lambda}_{min}$. It is necessary to prove that $\widetilde{\lambda}_{min} = \widehat{\lambda}_{min}$. Assume, for the purposes of contradiction, that $\widetilde{\lambda}_{min}$ is not the maximum in question. Therefore,

$$\exists N \mid \lambda_{min}^{(N)} \geq \widetilde{\lambda}_{min}$$

which, in turn

$$\Rightarrow \lambda_{min}^{(n)} \geq \widetilde{\lambda}_{min} \qquad \forall n \geq N$$

Remove the finite sequence $\{\lambda_{min}^{(n)}\}_{n=1}^{N}$ from the infinite sequence $\{\lambda_{min}^{(n)}\}$. This leaves the (still infinite) sequence $\{\lambda_{min}^{(n)}\}_{n=N+1}^{\infty}$. The definition of an accumulation point dictates that this removal of a finite sequence of elements will not affect $\widetilde{\lambda}_{min}$'s status as an accumulation point of the infinite sequence that remains. Let $\aleph(\widetilde{\lambda}_{min}, \delta)$ denote

the interval centred at $\tilde{\lambda}_{min}$ with a radius of $\|\tilde{\lambda}_{min} - \lambda_{min}^{(N)}\| \leq \delta$ as the *δ-neighbourhood* of $\tilde{\lambda}_{min}$ Now since $\{\lambda_{min}^{(n)}\}_{n=1}^{N}$ is monotone

$$\Rightarrow \aleph(\tilde{\lambda}_{min}, \delta) = \emptyset$$

i e , all points in the δ neighbourhood of $\tilde{\lambda}_{min}$ will have been removed This is a contradiction of $\tilde{\lambda}_{min}$ being an accumulation point Hence

$$\lambda_{min}^{(n)} \leq \tilde{\lambda}_{min} \qquad \forall n$$

As a corollary, it can be seen that since there can be no other λ_{min} in the interval $[\tilde{\lambda}_{min}, \hat{\lambda}_{min}]$ then

$$\tilde{\lambda}_{min} = \hat{\lambda}_{min}$$

□

A similar approach shows that the bounded monotone sequence $\{\lambda_{LP}^{(n)}\}_{n=1}^{\infty}$ will converge to $\hat{\lambda}_{LP}$ It is now possible to prove that no gap exists between $\hat{\lambda}_{min}$ and $\hat{\lambda}_{LP}$ Proceeding by contradiction assume that $\hat{\lambda}_{min} < \hat{\lambda}_{LP}$ Now for some suitable maximising support hyperplane $\hat{d} \in B_{\infty}$,

$$\hat{\lambda}_{min} = \lambda_{min}(\sum_{k=1}^{m} \hat{d}_k A_k)$$

$$\Rightarrow \hat{\lambda}_{min} = \min_{x} x^*(\sum_{k=1}^{m} \hat{d}_k A_k)x \qquad \forall x \in \partial B_2$$

Thus, there exists a certain $\hat{w} \in Co(W)$ for which

$$\hat{\lambda}_{min} = < \hat{d}, \hat{w} > \leq \|\hat{d}\|_{\infty}\|\hat{w}\|_1 = \|\hat{w}\|_1$$

Noting that any vector $\hat{w} \in Co(W)$ must provide an upper bound on $\hat{\lambda}_{LP}$, this yields the desired contradiction Therefore, it can be concluded that the new algorithm converges to $c_1(\alpha)$

3.8 Summary

A new algorithm for the computation of μ_{co} has been presented The new algorithm applies the Hahn-Banach theorem and uses a linear program solver at the heart of a

1-norm dual optimisation strategy Convergence has been demonstrated for the basic form of the algorithm

Several improvements to the basic algorithm have been described These improvements provide new bounds on μ_{co} which improve on the binary search approach used by the basic algorithm These new bounds can be divided into rigorous, so called hard, upper bounds on μ_{co} and soft candidate bounds The hard bounds are classified as those which arise from matrix pencil theory and the use of singular values The former is applicable only to the case of complex uncertainty while the latter is equally applicable to mixed uncertainty problems Soft bounds are deduced from the analysis of constraints that produce vectors close to the boundary of the convex set in question

The biggest single improvement that can be made to the basic algorithm is the reuse of good constraints in subsequent inner loop iterations A significant reduction in software overhead is possible if a sensible constraint pruning strategy, based on an analysis of these good constraints, is adopted The concept of whether a vector is close to the boundary and thus whether a constraint is tight, is a dynamic one which needs to be updated as the algorithm progresses Judicious pruning controls the size of a linear programming tableau and also improves computing times

Convergence of the improved form of the new algorithm carries over, since the convergence properties of the inner loop are unchanged by any of these amendments Since, at worst, a binary search can still be used for the outer loop it can be concluded that the improved form of the basic algorithm also converges to μ_{co}

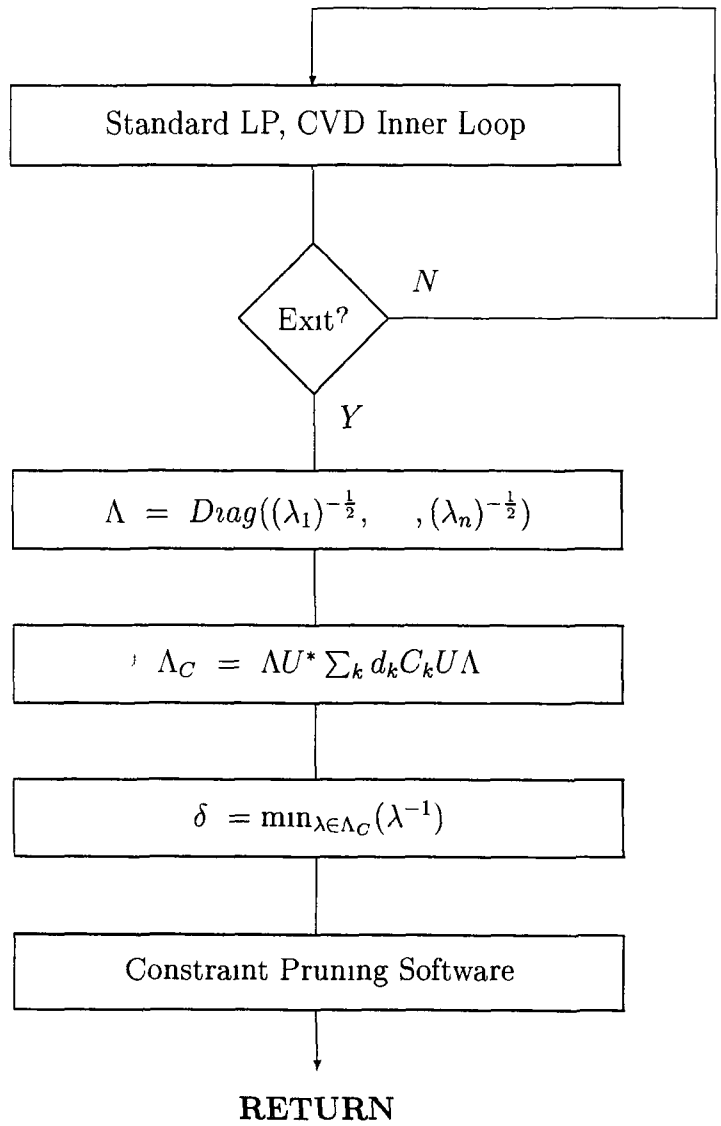


Figure 3 6 Flow Diagram for the Matrix Pencil Additions to the Basic Algorithm

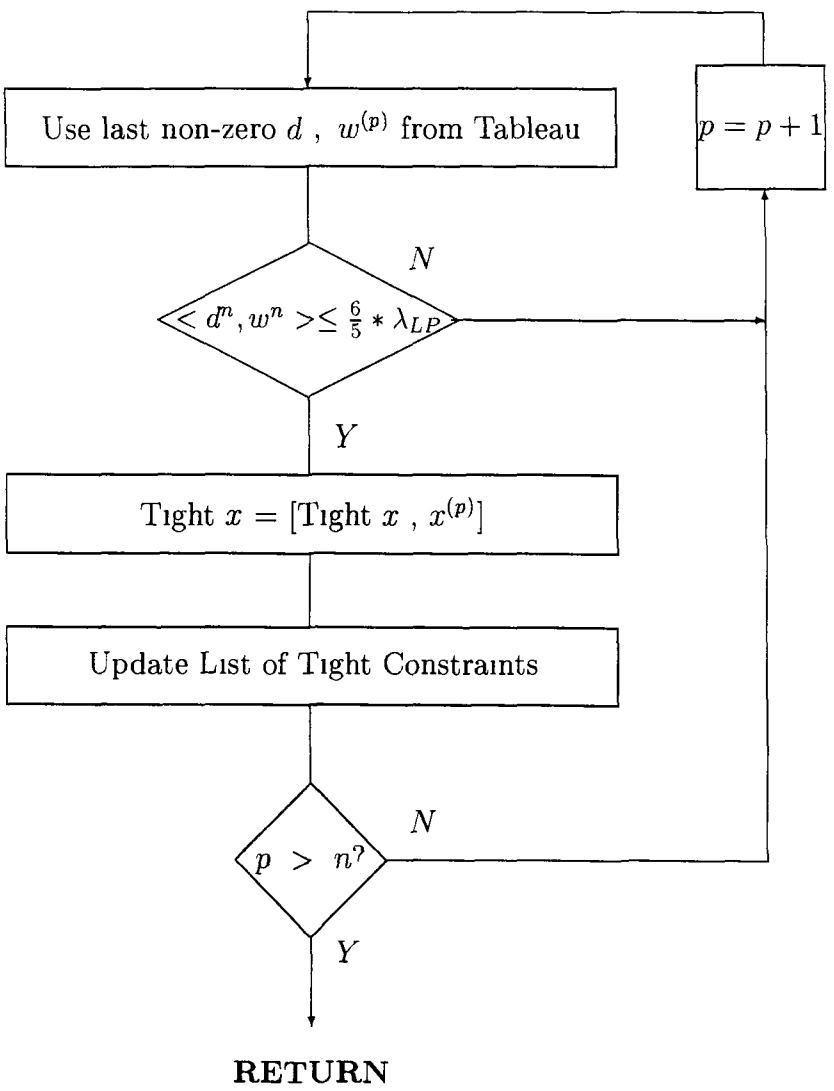


Figure 3 7 Flow Diagram for Constraint Pruning Additions to the Basic Algorithm

Chapter 4

Algorithm Performance

A significant amount of computational work has been carried out using the new algorithm developed in the previous chapter The aims of the first half of this chapter are to illustrate its effectiveness in a number of different areas These are -

1 **Validation** The algorithm can successfully calculate μ_{co} when faced with an arbitrary matrix

2 **Flexibility** The new algorithm has the capability to deal with a variety of different uncertainty structures and performance questions

3 **Comparison** The new algorithm should be competitive with existing commercially available software packages

The second half of this chapter justifies this decision to select a 1-norm dual optimisation strategy for the application of the Hahn-Banach Theorem This is achieved by comparing this strategy with other possible approaches, i e , 2-norm based methods This chapter also considers possible reductions to the amount of time spent inside an LP solver, which is a significant limiting factor on algorithm performance

4.1 Reliability of the Basic Algorithm

The first task is to demonstrate that the new algorithm works correctly

4.1.1 Complex Uncertainty

The new algorithm was tested on groups of 50 randomly generated $n \times n$ matrices, where $n \in \{3, \quad, 10\}$, amounting to 400 matrices in all Initially, complex scalar uncertainty was used Each matrix was pre-conditioned using Osborne's method The new algorithm was terminated when bounds on μ_{co} had been calculated to within 10% of each other For each matrix, software from the Multivariable Frequency Domain **(MFD)** Toolbox due to Ford *et al* [Ford 90] was also used to calculate an upper bound on μ_{co}

For all problem matrices the new method successfully produces bounds either side of the bound generated by the MFD code Table 4 1 illustrates a selection of the results for the 10×10 case

4.1.2 Real/Mixed Uncertainty

Next, the algorithm was tested using mixed/real uncertainty structures The test consisted of seven groups of 20 randomly generated $n \times n$ matrices, where $n \in \{3, \quad, 9\}$, amounting to 140 matrices in all In each case the matrices were scaled so as that μ_{co} for complex scalar uncertainty was equal to one correct to four decimal places The first uncertain parameter is complex whilst the others are constrained to be real This choice of mixed perturbation set is motivated by the practical example, (to come in the next chapter), where robust performance information is required when real parameters are allowed to vary Clearly, this results in a reduction in μ_{co} Again, the bounds were calculated to within 10% using the new method The commercially available μ**Tools** [Balas 94] package was also used on each problem

Table 4.1 Comparison of bounds for Complex μ_{co} on Random 10×10 Problems

MFD	*New Algorithm*	
	Upper	Lower
4.5477780e+01	4.6949562e+01	4.4713868e+01
4.0735292e+01	4.1628459e+01	3.9646152e+01
4.2691216e+01	4.4521960e+01	4.2401867e+01
3.9937359e+01	4.0759074e+01	3.8818166e+01
4.0991701e+01	4.2691438e+01	4.0658512e+01
3.7832464e+01	3.9045092e+01	3.7185802e+01
3.9598388e+01	4.1565733e+01	3.9586412e+01
4.1936080e+01	4.2882196e+01	4.0840187e+01
4.1948770e+01	4.2660314e+01	4.0628871e+01
3.8056896e+01	3.9108405e+01	3.7246100e+01
4.2501650e+01	4.3296810e+01	4.1235057e+01
4.0853072e+01	4.2084179e+01	4.0080170e+01
4.2081735e+01	4.2514627e+01	4.0490121e+01
4.0827317e+01	4.0868582e+01	3.8922459e+01
3.8852742e+01	3.9301678e+01	3.7430170e+01
3.8280309e+01	3.8542513e+01	3.6707155e+01
4.2787983e+01	4.3011945e+01	4.0963758e+01
4.2541183e+01	4.4136459e+01	4.2034723e+01
4.1917550e+01	4.2821906e+01	4.0782767e+01
3.8156433e+01	3.9399928e+01	3.7523741e+01
4.4306751e+01	4.5624946e+01	4.3452330e+01
4.3219731e+01	4.3870193e+01	4.1781136e+01
4.2441299e+01	4.3390625e+01	4.1324405e+01
4.1256513e+01	4.3155041e+01	4.1100039e+01
4.2494935e+01	4.4091988e+01	4.1992370e+01
4.1423625e+01	4.2706359e+01	4.0672723e+01
4.0089373e+01	4.1058700e+01	3.9103524e+01

Table 4 2 illustrates, using the 9×9 case, how the two sets of bounds complement each other The bounds are mutually consistent in every case This verifies that both suites of software are calculating the same bound on μ It should also be noted from Table 4 2 that the μTools software produces bounds in the mixed uncertainty case that can be a significant distance from each other It is not possible to instruct the μTools software to return with bounds that are within a specified distance of each other

Table 4.2 Comparison of bounds for Mixed μ_{co} on Random 9×9 Problems

μTools		New Algorithm	
Upper	*Lower*	*Upper*	*Lower*
9 5180946e-01	9 4404890e-01	9 4418788e-01	9 4379956e-01
8 6362178e-01	8 3319262e-01	8 3714047e-01	8 3612690e-01
9 4226577e-01	7 7905954e-01	9 2928673e-01	9 2886588e-01
5 0757599e-01	7 7454764e-02	5 0551365e-01	5 0508452e-01
6 0486305e-01	5 1724870e-01	5 8385521e-01	5 8384955e-01
9 2431295e-01	9 2054541e-01	9 2065533e-01	9 2037948e-01
7 5357929e-01	3 6771778e-01	7 3647304e-01	7 3457545e-01
6 1827381e-01	6 1756959e-01	6 1782457e-01	6 1782303e-01
6 9627845e-01	6 8596537e-01	6 8615863e-01	6 8585786e-01
6 8340370e-01	6 4391253e-01	6 4802947e-01	6 4774501e-01
7 3598597e-01	6 9301243e-01	7 3585870e-01	7 3582907e-01
8 8772607e-01	8 8515638e-01	8 8520088e-01	8 8519940e-01
7 2130193e-01	4 9271686e-01	6 8280543e-01	6 8258569e-01
8 8330906e-01	8 2031115e-01	8 7249073e-01	8 7242408e-01
9 4461395e-01	9 3386788e-01	9 3395356e-01	9 3376206e-01
8 4652827e-01	8 2070436e-01	8 3898033e-01	8 3870618e-01
9 7194397e-01	9 1093383e-01	9 2742062e-01	9 2682111e-01

4.1.3 Effect of Problem Size on Computing Times

This subsection considers the increased computing time required to generate bounds within a given distance of each other when problem size is increased The number of floating point operations, *flops*, required to arrive at the bounds of Table 4 1 in subsection 4 1 1 were as follows

Table 4.3 FLOPS Required Vs. Problem Size
for Complex μ_{co} calculation

	MFD		New Algorithm	
N	*Average*	*Worst*	*Average*	*Worst*
3	4 1391640e+04	7 5077000e+04	1 6223940e+04	4 8838000e+04
4	6 8252520e+05	7 5706980e+06	8 1368620e+04	2 7788100e+05
5	2 5701669e+06	2 1239972e+07	2 1997250e+05	6 1101700e+05
6	5 1444434e+06	3 7661396e+07	5 3629888e+05	2 4621900e+06
7	1 3604973e+07	4 8709478e+07	1 2911863e+06	4 9182090e+06
8	3 3935867e+07	8 4697246e+07	2 0061604e+06	4 8031250e+06
9	4 5495608e+07	1 1293525e+08	3 6704778e+06	1 2405485e+07
10	9 1166564e+07	1 5248893e+08	5 3119635e+06	1 4353530e+07

Table 4 3 shows the average and worst case flops required by the new algorithm to determine the bounds on μ_{co} for complex only uncertainty For comparison purposes, the flops required by the MFD algorithm are also listed Note that the two algorithms have different termination conditions so rigorous speed comparisons cannot be made using this data It should be noted that the upper bound produced by the MFD code will generally be much closer to μ_{co} than the upper bound generated by the basic form of the new algorithm, when both are given the same computing time

In a similar fashion, Table 4 4 presents average and worst case flops required by the new algorithm and μTools to calculate bounds on μ_{co} for the mixed/real case The first thing to note is that mixed uncertainty problems require a significantly greater computational effort than a purely complex problem of similar size. In this case, the new algorithm requires an order of magnitude more flops to generate bounds within 10% of each other

Table 4.4 FLOPS Required Vs. Problem Size for Mixed μ_{co} Calculation				
	μ Tools		New Algorithm	
N	*Average*	*Worst*	*Average*	*Worst*
3	7 7940950e+04	1 0438900e+05	2 2811610e+05	5 3049100e+05
4	1 6178200e+05	2 4652500e+05	1 1284165e+06	2 9312970e+06
5	2 6486980e+05	3 4461800e+05	2 9537872e+06	6 1191670e+06
6	1 6178200e+05	2 4652500e+05	1 1284165e+06	2 9312970e+06
7	6 5679335e+05	7 7821400e+05	1 6604177e+07	3 3623160e+07
8	9 4500340e+05	1 2021720e+06	5 1820302e+07	2 5860051e+08
9	1 3624298e+06	2 0719260e+06	5 7217077e+07	1 3064019e+08

The exponential nature of the growth in computing times can be clearly seen if one graphs the results obtained as in Fig 4 1 The continuous line is the average flops required whilst the broken line represents the worst case problem that was encountered It can be seen that computing times for the new algorithm compare favourably with μTools on low order mixed uncertainty problems However, the new algorithm begins to exhibit problems when matrix size begins to exceed $n = 7$

4.2 Improvements to the Basic Algorithm

To appreciate the benefits of the suggested improvements outlined in the previous chapter, particularly with smaller problem sizes, the accuracy of the bounds generated must be taken into account Significant improvements to the speed of the algorithm can only be judged in this context For example, computing times cannot be significantly reduced for the problems presented in the last section, if bounds within 10% of each other are all that are required This is due to the significant software overhead required in solving the extra linear programming problems that are a feature of the improvements Also, it should be stated at the outset that, despite the improvements, existing commercial code for the computation of μ_{co} will, in general, deliver bounds faster than the new algorithm

However, the new algorithm does possess some key advantages over existing strategies

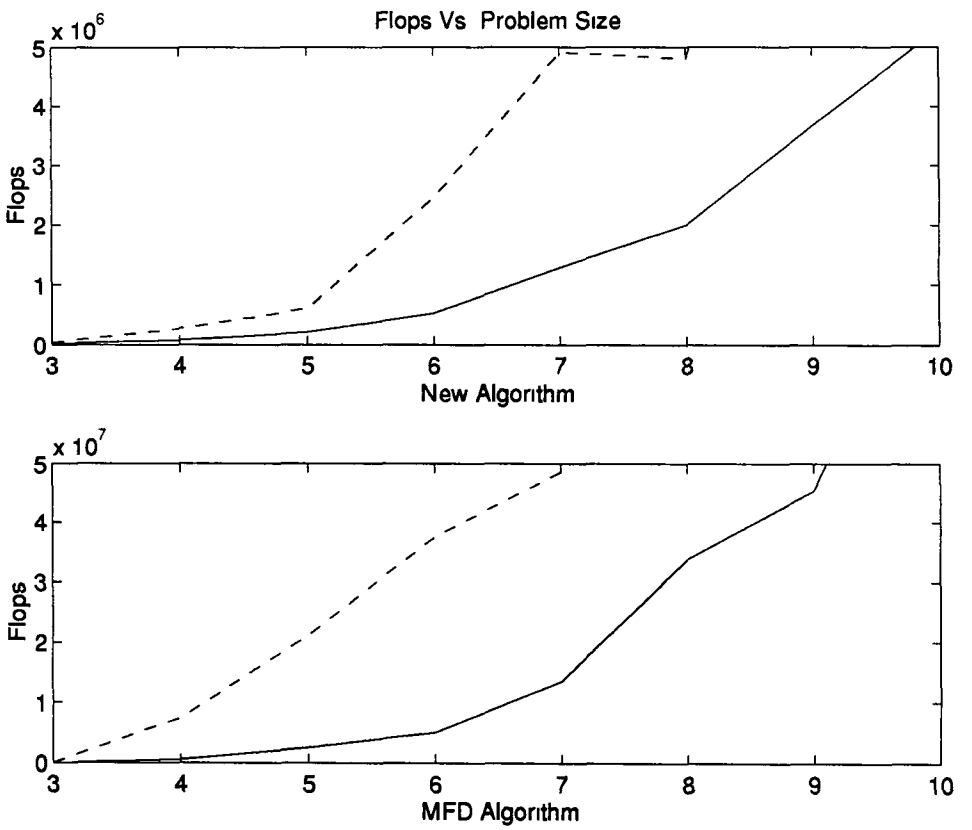


Figure 4 1 Time Required by New Algorithm (Top) and MFD Code (Bottom) to Compute Bounds Within 10% of μ_{co} Against Increasing Problem Size

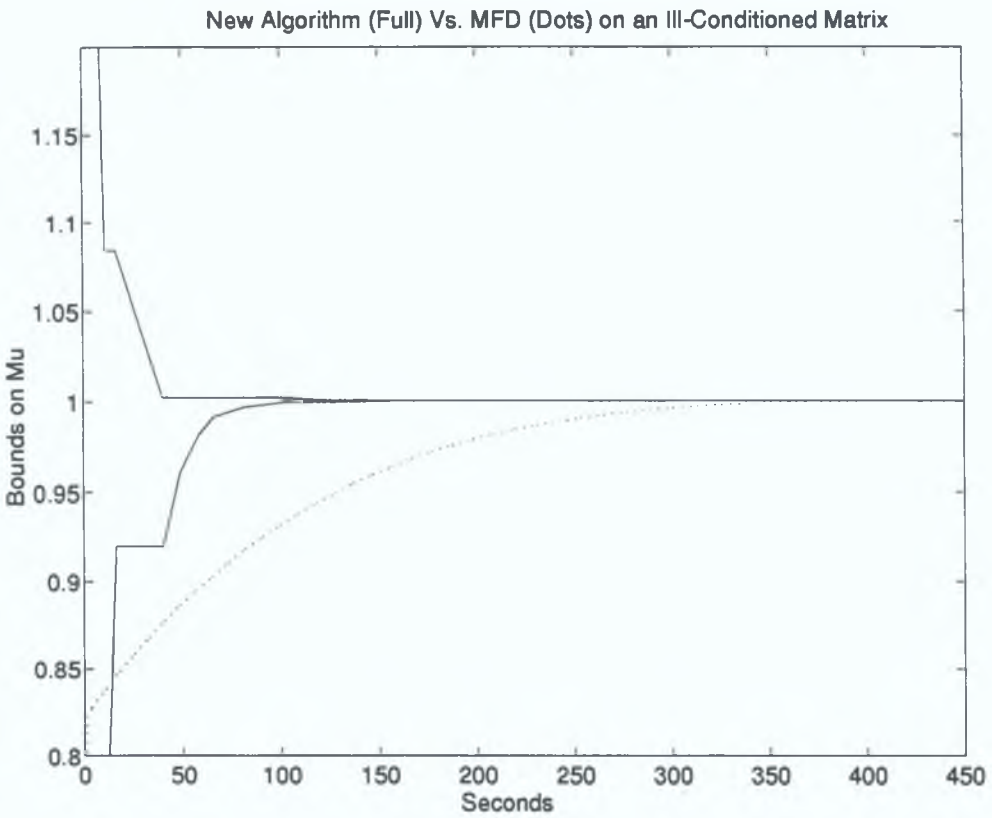


Figure 4.2: Time Required by New Algorithm (Full) and MFD (Dot) to Compute bounds on μ_{co} for a particular ill-conditioned matrix.

It is more reliable than the MFD Toolbox code. It consistently offers tight upper and lower bounds on problem matrices with purely complex or mixed uncertainty structures. This is particularly evident when the matrix in question is ill-conditioned. The new algorithm offers consistently better mixed μ bounds than those offered by the μTools software.

4.2.1 Reliability

Consider the ill-conditioned benchmark 3×3 problem presented in [Doyle 82]. This matrix maybe rescaled so as that $\mu_{co} = 1$ correct to four decimal places. It is clear from Fig. 4.2 that the MFD code (dotted line) has problems getting close to μ_{co}. In contrast, it can be seen how the New Algorithm (full line), calculates tight bounds in a fraction of the time. Extensive numerical experience has failed to isolate a problem where the new algorithm experiences difficulty in locating tight bounds on μ_{co}.

4.2.2 Accuracy

Existing commercial algorithms that use gradient methods to compute μ_{co} necessarily suffer performance problems during later iterations Reductions in speed and "stitching", where little improvement is noted for a number of iterations, has been widely reported [Fan 91] For example, this can be observed when a more accurate upper bound is requested from the μTools software Numerical experience indicates that the new algorithm does not suffer from these problems

Take, for example, the 3×3 problem just presented in the previous subsection Computation speed depends on how quickly the new algorithm can determine whether a certain α is a valid upper or lower bound on $k_{m_{co}} = \mu_{co}^{-1}$ Two values of α are chosen One is less than $k_{m_{co}}$ whilst one is greater However, both are correct to five significant figures The graph in Fig 4 3 illustrates the progress of the upper bound on $c_1(\alpha)$ through to the conclusion of the inner loop, where $\alpha > k_{m_{co}}$ Fig 4 4 repeats this process for $\alpha < k_{m_{co}}$ The figures show how the algorithm correctly recognises that the two candidate α's are above and below $k_{m_{co}}$ respectively No stitching is noticed In both cases the algorithm progresses to the correct answer at a rate which is better than linear This behaviour has also been observed on arbitrary problem matrices with complex and mixed uncertainty structures

For this benchmark 3×3 problem, bounds on μ_{co} have been calculated accurately to twelve decimal places At each iteration of the inner loop the progression to an answer was better than linear For the record, the bounds on μ_{co} for the problem in [Doyle 82], rescaled by $\frac{1}{10\,21}$, correct to 12 decimals are -

$$\mu_{co} \in (0\,99980166004654, \quad , 0\,99980166004658)$$

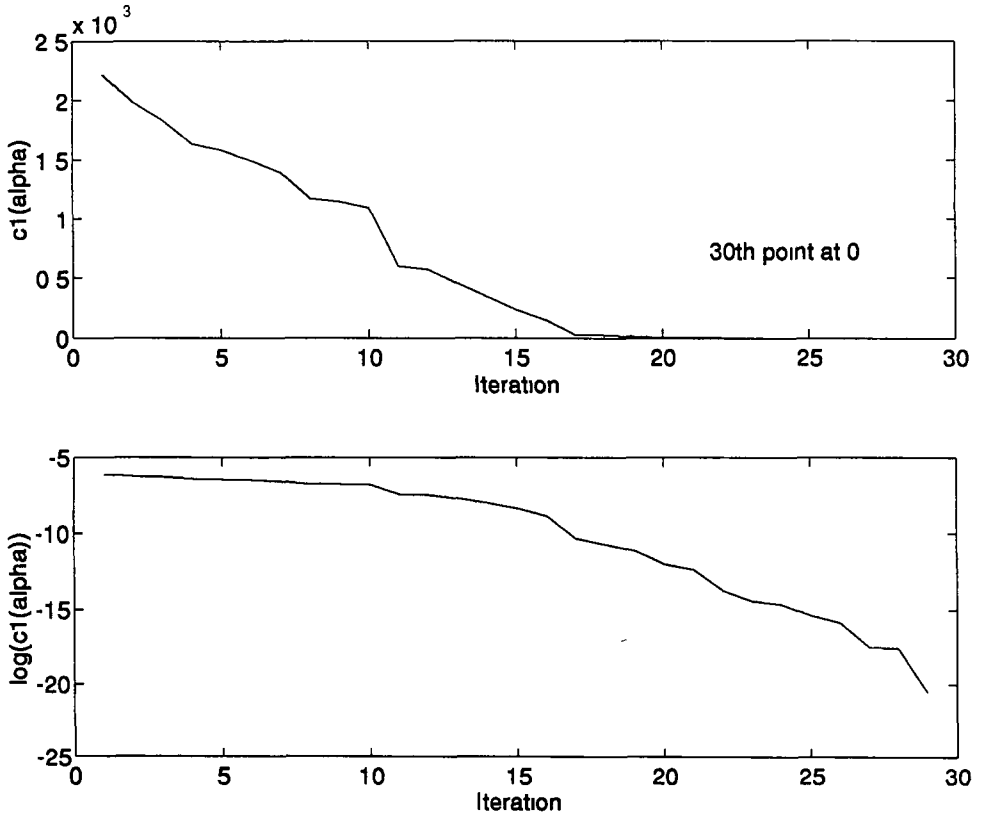


Figure 4 3 Progress of Upper Bound on $c_1(\alpha)$ when $\alpha > k_{m_{co}}$

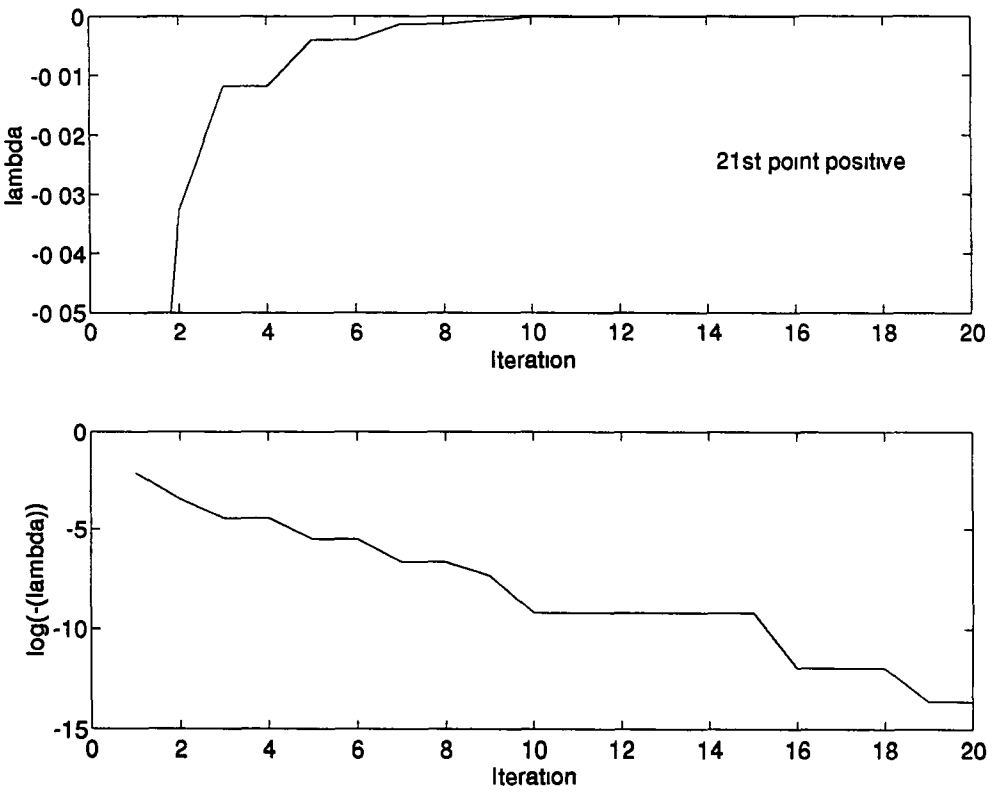


Figure 4 4 Progress of Lower Bound on $c_1(\alpha)$ when $\alpha < k_{m_{co}}$

To compare the accuracy of the bounds generated by the new algorithm with existing commercial code and to provide further validation, 20 pseudo-random matrices with $\mu_{co} = 1$ were generated using a construction due to Fan [Fan:2 86]. Table 4.5 lists the bounds obtained by the New Algorithm alongside those due to μTools and the MFD Toolbox. Notice the excellent agreement between the bounds provided by all three algorithms. All, in general, produce tight bounds on μ_{co}. However the MFD code does stitch on two occasions. Table 4.6 lists the number of flops required by each method to arrive at these bounds.

Table 4.5 Comparison of Tight Bounds for Complex μ_{co}

MFD	*μ Tools*		*New Algorithm*	
	Upper	*Lower / Power*	*Upper*	*Lower*
1.0000000e+00	1.0009313e+00	1.0000000e+00	1.0000000e+00	1.0000000e+00
1.0000000e+00	1.0013121e+00	1.0000000e+00	1.0000000e+00	1.0000000e+00
1.0000000e+00	1.0003828e+00	1.0000000e+00	1.0000000e+00	1.0000000e+00
1.0035227e+00	1.0036257e+00	1.0000000e+00	1.0000000e+00	1.0000000e+00
1.0000000e+00	1.0092200e+00	1.0000000e+00	1.0000000e+00	1.0000000e+00
1.0000001e+00	1.0007105e+00	1.0000000e+00	1.0000000e+00	1.0000000e+00
1.0000001e+00	1.0005578e+00	1.0000000e+00	1.0000000e+00	1.0000000e+00
1.0000000e+00	1.0007517e+00	1.0000000e+00	1.0000000e+00	1.0000000e+00
1.0000000e+00	1.0022791e+00	1.0000000e+00	1.0000000e+00	1.0000000e+00
1.0000000e+00	1.0000004e+00	1.0000000e+00	1.0000002e+00	9.9999988e-01
1.0000000e+00	1.0007049e+00	1.0000000e+00	1.0000000e+00	9.9999999e-01
1.0000001e+00	1.0000088e+00	1.0000000e+00	1.0000000e+00	1.0000000e+00
1.0000001e+00	1.0005955e+00	1.0000000e+00	1.0000000e+00	1.0000000e+00
1.0000001e+00	1.0011454e+00	1.0000000e+00	1.0000000e+00	1.0000000e+00
1.0000000e+00	1.0003084e+00	1.0000000e+00	1.0000000e+00	1.0000000e+00
1.0080876e+00	1.0039969e+00	1.0000000e+00	1.0000000e+00	1.0000000e+00
1.0000000e+00	1.0009142e+00	1.0000000e+00	1.0000000e+00	1.0000000e+00
1.0000000e+00	1.0002782e+00	1.0000000e+00	1.0000000e+00	1.0000000e+00
1.0000000e+00	1.0000720e+00	1.0000000e+00	1.0000000e+00	1.0000000e+00
1.0000000e+00	1.0034524e+00	1.0000000e+00	1.0000000e+00	1.0000000e+00

Table 4.6 Flops Required for Tight Bounds on Complex μ_{co}

MFD	*μ Tools*	*New Algorithm*
7.8815000e+04	3.8101000e+04	6.6114230e+06
5.6398000e+04	2.1649000e+04	3.9717730e+06
2.7517000e+04	4.1194000e+04	5.2260150e+06
5.1762400e+05	1.7352000e+04	3.6287500e+06
7.2450000e+04	4.3118000e+04	3.5454010e+06
6.9285000e+04	1.7856000e+04	2.8232370e+06
1.2511300e+05	3.4130000e+04	6.0607740e+06
2.6346000e+04	3.6624000e+04	1.1420925e+07
5.1917000e+04	1.9946000e+04	3.2802370e+06
1.9148500e+05	3.6200000e+04	3.7046840e+06
5.2787000e+04	2.9341000e+04	2.9231390e+06
7.8744000e+04	2.8555000e+04	3.5717770e+06
1.5524600e+05	2.0099000e+04	4.3156270e+06
8.2768000e+04	1.8493000e+04	2.9882630e+06
4.8451000e+04	3.6101000e+04	6.6144560e+06
6.4854400e+05	1.9341000e+04	3.5399170e+06
3.6440000e+04	2.0577000e+04	4.3991600e+06
2.6969000e+04	3.5193000e+04	3.8425680e+06
3.4511000e+04	1.6374000e+04	2.8145150e+06
5.2955000e+04	1.7227000e+04	3.7489020e+06

Clearly, the new algorithm has spent a lot longer at the problems. However, it is not possible to allow the commercial packages to run until bounds with a prescribed level of accuracy have been achieved. Thus, the fairest means of algorithm comparison can be a difficult question. The gradient method used by both μTools and the MFD Toolbox for the upper bound indicates that further improvement in their respective upper bounds will be slow. This is particularly noticeable for mixed μ. For example, any improvements to the μTools bounds of Table 4.2 are computationally expensive to obtain.

Improved Accuracy Vs. Flops Requirement

The basic form of the new algorithm uses a simple binary search between the upper and lower bounds on μ_{co} to yield an extra place of decimals per three inner iterations. It is difficult to produce a similar rule of thumb for the improved form of the algorithm. The improvement per iteration is no longer a constant for any given matrix. Typical values of bounds versus flops required after each iteration are presented in Table 4.7 for a sample problem generated using Fan's construction [Fan:2 86].

Table 4.7 Sample Bounds Vs.Flops Required for Complex μ_{co}

Lower	*Upper*	*Flops*
9.9988246e-01	1.0023858e+00	5.7120100e+05
9.9998417e-01	1.0000389e+00	4.9664900e+05
9.9999957e-01	1.0000026e+00	6.1309600e+05
9.9999999e-01	1.0000000e+00	1.0016250e+06
9.9999999e-01	1.0000000e+00	6.0996000e+05
1.0000000e+00	1.0000000e+00	1.4832960e+06

This table demonstrates the behaviour of the algorithm that has been observed in the majority of cases. The gap between the upper and lower bound is only being bisected during iterations 4 to 6. This suggests that the improvements to the new algorithm are having little or no effect at this point. Therefore, the significant software overhead required to exactly calculate $c_1(\alpha)$ during the latter stages may not be a judicious use of computing time. Consequently the flop count for the new algorithm in Table 4.6 should be considered conservative.

Remark - On Computing Times with Mixed Uncertainty

For the mixed case, it takes significantly longer to determine bounds of similar accuracy to those which can be attained with complex only uncertainty. This is not

unexpected The absence of bounds due to matrix pencil theory and "tight" constraints is a factor Also, the geometric extension from the complex only to the mixed uncertainty case is not straightforward Indeed, the geometry of the mixed case leads to vectors with many component zeros, thus causing an accumulation of vectors close to the null vector This results in the need for a significantly greater number of iterations of the inner loop to determine comparable bounds on mixed μ_{co}

4.2.3 Mixed/Real Uncertainty

Next, the effect of the singular value bound for mixed uncertainty problems which was discussed in Chapter 3 is considered This is a rigorous lower bound on mixed μ_{co} Table 4 8 illustrates the effectiveness of this bound Note that an improved bound is only determined when the seed α is larger than μ_{co}

Table 4.8 Effect of Singular Value Bound on Calculation of Mixed μ_{co}

Iteration	$Upper^{(n)}$	$Lower^{(n)}$	Standard $\alpha^{(n)}$	Improved $Lower^{(n+1)}$
1	0 673419054	0 448946036	0 538735243	-
2	0 673419054	0 538735243	0 598594714	-
3	0 673419054	0 598594714	0 633806168	-
4	0 673419054	0 633806168	0 653012416	-
5	0 673419054	0 653012416	0 663058761	0 660520606
6	0 660520606	0 653012416	0 656745052	0 655979658
7	0 655979658	0 653012416	0 654492674	0 654336575
8	0 654336575	0 653012416	0 653673825	-
9	0 654336575	0 653673825	0 654005032	0 653979405
10	0 653979405	0 653673825	0 653826579	-
11	0 653979405	0 653826579	0 653902983	-

Clearly, the improvement is not dramatic However the bound does allow the new algorithm to do better than a binary search for mixed uncertainty problems

Remark - On Flexibility with Mixed Uncertainty

It is a straightforward process with the new algorithm to specify that the rescaling or gain parameter α only acts on certain blocks of a given problem matrix There is no facility to do this with the μTools software The requirement to be able to do this can occur quite naturally in practice For example, when the effect of component uncertainty on filter performance is analysed later on, it will be seen that it is essential that the real blocks representing the component uncertainty are independent of α

4.3 1-Norm Vs. 2-Norm Methods

The aim of this section is to justify the use of a 1-norm dual optimisation strategy over its 2-norm primal or dual counterpart The focus will be on two key areas
(i) The proximity problem which is at the heart of the numerical range formulation
(ii) Which method produces better bounds on μ_{co} in an equivalent time
Whilst this section achieves this aim, by relaxing accuracy requirements, certain situations can occur which tend to favour either of the 2-norm methods For completeness these situations are briefly mentioned Throughout this section, the effect of possible improvements to the basic form of the algorithm in question is not considered

4.3.1 The Proximity Problem

The objective of this section is to show that a 1-norm dual method is the best way to determine whether $0 \in Co(W(\alpha))$ Initially its performance is compared with a 2-norm primal method in a standardised environment Again, consider the 3×3 problem presented in [Doyle 82] Complex uncertainty is used Two seed values of α_1, α_2 are chosen

$$\alpha_1 \geq k_{m_{co}} \Rightarrow 0 \in Co(W(\alpha_1))$$

$$\alpha_2 \leq k_{m_{co}} \Rightarrow 0 \notin Co(W(\alpha_2))$$

However both α_1 and α_2 are equal to $k_{m_{co}}$ correct to five significant figures The progress of the upper and lower bounds on $c(\alpha)$ for either method is recorded for both candidate α's This progress is presented on linear and logarithmic axes in

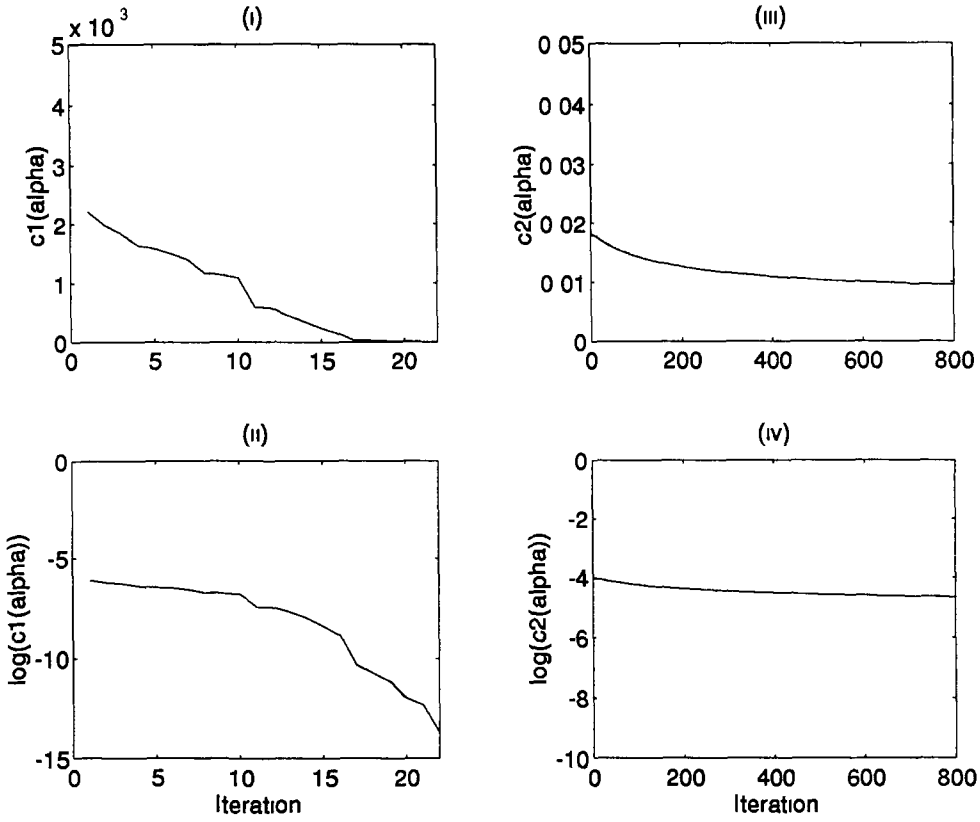


Figure 4 5 Comparison of 1-norm dual (i),(ii) with 2-norm primal (iii), (iv) when verifying that $0 \in Co(W)$ for $\alpha > k_{m_{co}}$

Figs 4 5 and 4 6 Separate axes have been used to highlight the difference between the two norms used to generate $c_1(\alpha)$ and $c_2(\alpha)$

The 1-norm dual approach converges to an answer more quickly than the 2-norm primal method Indeed for both α_1 and α_2 the 2-norm approach will fail to answer the proximity question in any reasonable amount of time Encouragingly, the 1-norm dual method seems to offer only modest growth in the number of iterations required with increasing accuracy The log plots that are presented in the figures suggest that these curves follow a trend which is roughly linear or better This motivates the conjecture that a 1-norm dual method offers an exponential rate of convergence for computing $c_1(\alpha)$ 2-norm primal methods, in common with many gradient based optimisation strategies, suffer from rapid growth in the computation time required when α tends toward the actual value of $k_{m_{co}}$ Moreover, as higher levels of accuracy are required, 2-norm primal methods will require a much greater number of iterations (and thus time) for convergence For reasonable computing times and, given the finite

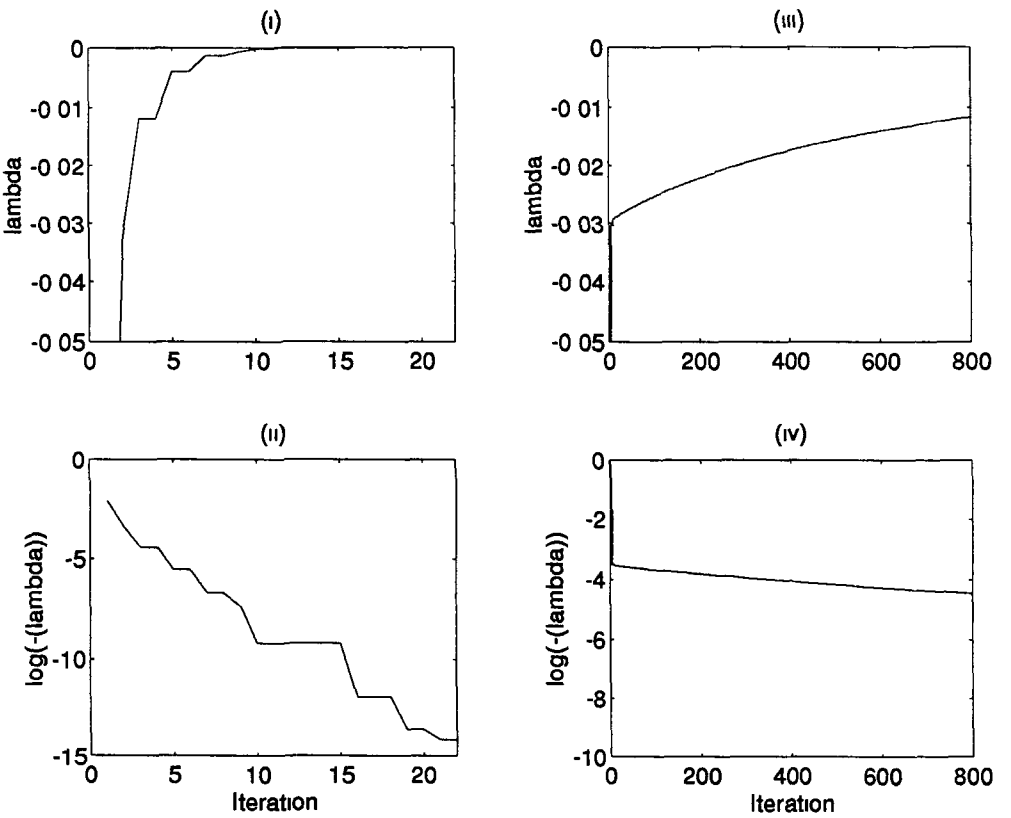


Figure 4 6 Comparison of 1-norm dual (i),(ii) with 2-norm primal (iii),(iv) when verifying that $0 \notin Co(W)$ for $\alpha < k_{m_{co}}$

precision of computer arithmetic, it is necessary to work with some arbitrary ϵ When $c_2(\alpha) \leq \epsilon$ one *assumes* that 0 will be inside $Co(W(\alpha))$

Consider the construction due to Fan [Fan 2 86], which produces psuedo random matrices with $\mu_{co} = 1$ Further, consider three candidate α's

$$1\ \alpha_i = 1 = k_{m_{co}} \Rightarrow 0 \in \partial Co(W(\alpha_i)) \qquad i\,e\ ,\ c(\alpha_i) = 0$$
$$2\ \alpha_{ii} = 1 + 10^{-14} \Rightarrow 0 \notin Co(W(\alpha_{ii}))$$
$$3\ \alpha_{iii} = 1 - 10^{-14} \Rightarrow 0 \in Co(W(\alpha_{iii}))$$

These α's are as close to each other as the operating precision of the standard DOS based version of *MATLAB* will allow The 1-norm dual, 2-norm primal and dual algorithms are applied to such matrices A maximum computing time of 200 seconds was allowed for each algorithm It should be noted that the 1-norm dual algorithm does not require anything like this amount of time to solve this problem Typical progress is presented on logarithmic axes in Fig 4 7 For ease of comparison, the upper and lower bounds on $c_1(\alpha)$ and $c_2(\alpha)$ are plotted on the same axes only in Fig 4 7 Strict equivalence of the distances should not be inferred as different norms are involved To underline the advantages of a 1-norm dual approach, it is instructive to look at the bounds stored by each algorithm during the last ten iterations, for each value of α, in tabular form

These tables clearly show how the 1-norm dual algorithm is sensitive to the smallest possible variations in α when detecting whether $c_1(\alpha) \neq 0$ The other algorithms fail to achieve this task

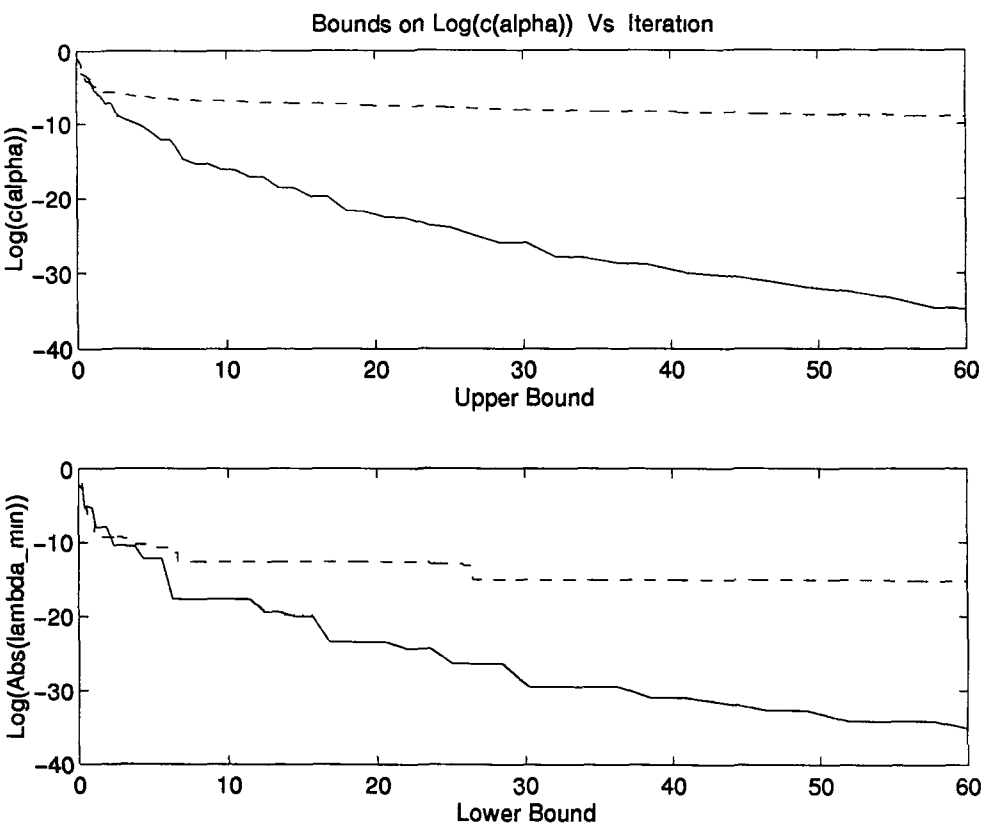


Figure 4 7 Bounds on $log(c(\alpha_i))$ using 1-norm dual (full), 2-norm primal (dash) and 2-norm dual (dot) methods $\alpha = k_{m_{co}}$

Table 4.9 Final Iterations of Each Algorithm for $\alpha_i = k_{m_{co}}$

1-norm Dual		2-norm Primal		2-norm Dual	
Lower	*Upper*	*Lower*	*Upper*	*Lower*	*Upper*
-1 6177e-13	3 6971e-13	-4 5060e-10	1 6048e-05	-2 4504e-09	1 1663e-04
-3 8056e-14	2 9447e-13	-4 5060e-10	1 6048e-05	-2 4504e-09	1 1663e-04
-3 8056e-14	1 0602e-13	-4 5060e-10	1 6048e-05	-2 4504e-09	1 1663e-04
-1 7266e-14	6 5854e-14	-4 5060e-10	1 6046e-05	-2 4504e-09	1 1663e-04
-6 7145e-15	3 4322e-14	-4 5060e-10	1 6045e-05	-2 4504e-09	1 1663e-04
-6 7145e-15	1 3610e-14	-4 5060e-10	1 6045e-05	-2 4504e-09	1 1663e-04
-1 6820e-15	8 5651e-15	-4 5060e-10	1 6045e-05	-2 4504e-09	1 1663e-04
-1 6820e-15	3 3401e-15	-4 5060e-10	1 6042e-05	-2 4504e-09	1 1663e-04
-1 5458e-15	9 5363e-16	-4 5060e-10	1 6041e-05	-2 4504e-09	1 1663e-04
-4 4351e-16	8 2161e-16	-4 5060e-10	1 6041e-05	-2 4504e-09	1 1663e-04
-4 4351e-16	0 0000e+00	-4 5060e-10	1 6041e-05	-2 4504e-09	1 1663e-04

Table 4.10 Final Iterations of each Algorithm for $\alpha_n = k_{m_{co}} + 10^{-14}$

1-norm Dual		2-norm Primal		2-norm Dual	
Lower	*Upper*	*Lower*	*Upper*	*Lower*	*Upper*
-1 6902e-13	5 1506e-12	-1 5930e-10	6 3161e-06	-4 0014e-07	7 4686e-04
-1 6902e-13	8 8553e-13	-1 5930e-10	6 3157e-06	-4 0014e-07	7 4686e-04
-1 6902e-13	7 4508e-13	-1 5930e-10	6 3156e-06	-4 0014e-07	7 4686e-04
-1 6902e-13	3 6241e-13	-1 5930e-10	6 3156e-06	-4 0014e-07	7 4686e-04
-4 5384e-14	2 8716e-13	-1 5930e-10	6 3156e-06	-4 0014e-07	7 4686e-04
-4 5384e-14	9 8799e-14	-1 5930e-10	6 3155e-06	-4 0014e-07	7 4686e-04
-2 4543e-14	5 8644e-14	-1 5930e-10	6 3155e-06	-4 0014e-07	7 4686e-04
-1 4084e-14	2 7062e-14	-1 5930e-10	6 3154e-06	-4 0014e-07	7 4686e-04
-1 4084e-14	6 3552e-15	-1 5930e-10	6 3154e-06	-4 0014e-07	7 4686e-04
-8 9619e-15	1 3099e-15	-1 5930e-10	6 3154e-06	-4 0014e-07	7 4686e-04
-8 9619e-15	0 0000e+00	-1 5930e-10	6 3154e-06	-4 0014e-07	7 4686e-04

Table 4.11 Final Iterations of each Algorithm for $\alpha_{ini} = k_{m_{co}} - 10^{-14}$

1-norm Dual		2-norm Primal		2-norm Dual	
Lower	*Upper*	*Lower*	*Upper*	*Lower*	*Upper*
-3 0764e-14	3 0174e-13	-1 5464e-08	3 1234e-05	-4 0014e-07	7 4686e-04
-3 0764e-14	1 1324e-13	-1 5464e-08	3 1234e-05	-4 0014e-07	7 4686e-04
-9 9878e-15	7 3143e-14	-1 5464e-08	3 1233e-05	-4 0014e-07	7 4686e-04
4 8962e-16	7 3143e-14	-1 5464e-08	3 1232e-05	-4 0014e-07	7 4686e-04
4 8962e-16	4 1586e-14	-1 5464e-08	3 1231e-05	-4 0014e-07	7 4686e-04
4 8962e-16	2 0958e-14	-1 5464e-08	3 1231e-05	-4 0014e-07	7 4686e-04
5 5302e-15	1 5821e-14	-1 5464e-08	3 1229e-05	-4 0014e-07	7 4686e-04
5 5302e-15	1 0639e-14	-1 5464e-08	3 1228e-05	-4 0014e-07	7 4686e-04
5 6176e-15	8 1499e-15	-1 5464e-08	3 1228e-05	-4 0014e-07	7 4686e-04
6 8691e-15	8 0738e-15	-1 5464e-08	3 1226e-05	-4 0014e-07	7 4686e-04
6 8691e-15	6 8383e-15	-1 5464e-08	3 1225e-05	-4 0014e-07	7 4686e-04

The 2-norm dual algorithm fails to make any distinction between the values of α that are either side of $k_{m_{co}}$ At these accuracy levels the author has yet to find an example where the 2-norm dual approach can produce correct answers despite using multiple different starting points In contrast, the final bounds of the 2-norm primal approach are sensitive to variations in α This demonstrates the algorithm's convergence properties However, the results indicate that an extremely long period of time may be necessary for this convergence to occur Bear in mind that the upper bound on $c_2(\alpha_{ii})$ still needs eight decimal places worth of improvement to arrive at an answer similar in precision to that provided by the 1-norm dual algorithm It is also unclear, *a priori*, whether the 2-norm primal upper bound on $c_2(\alpha_{ii})$ is "small enough" to be assured that $c_2(\alpha_{ii})$ does in fact equal zero

In Figs 4 8 and 4 9 the performance of the 1-norm dual algorithm is plotted The 2-norm primal and dual curves are similar to their counterparts in Fig 4 7 and are omitted for reasons of clarity The results plotted in the figures are typical for all the matrices that were analysed

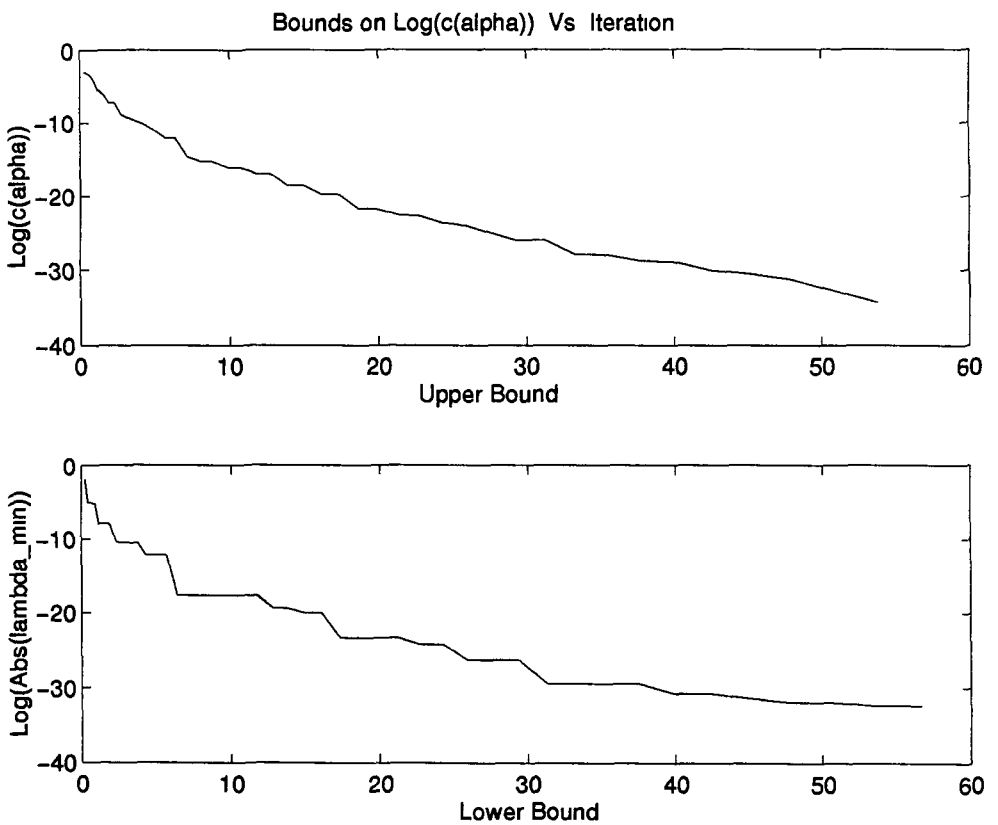


Figure 4 8 Bounds on $log(c_1(\alpha_u))$ using a 1-norm dual method $\alpha_u = k_{m_{co}} + 10^{-14}$

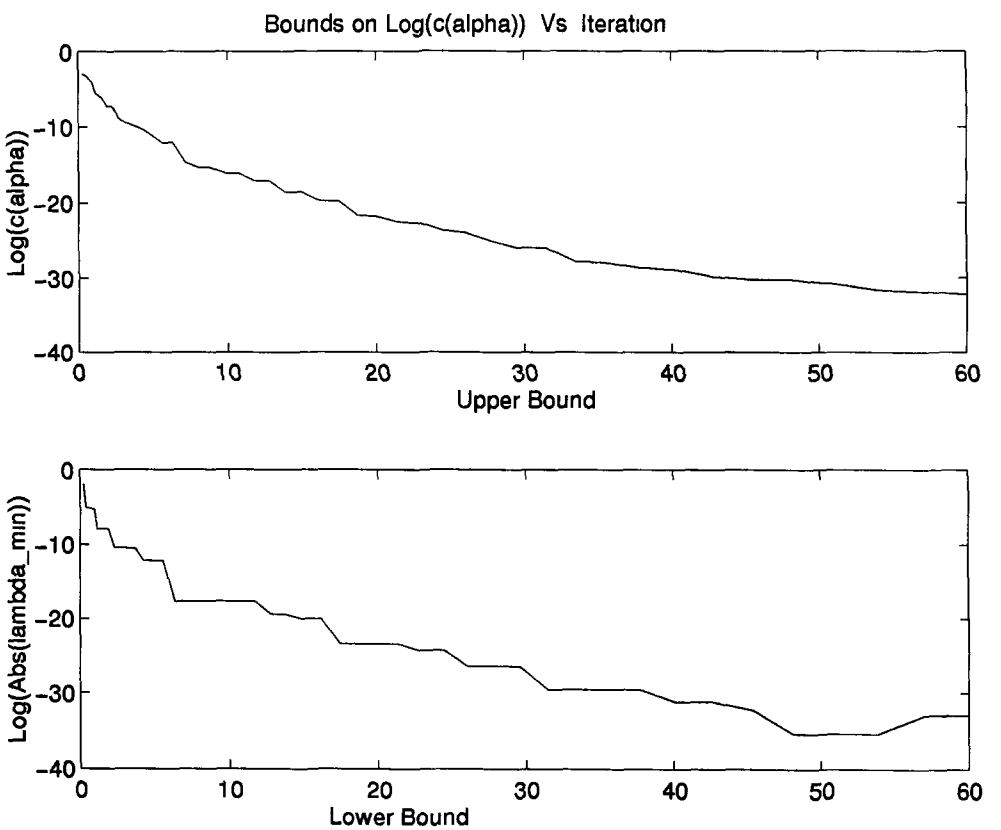


Figure 4 9 Bounds on $log(c_1(\alpha_{m}))$ using a 1-norm dual method $\alpha_{m} = k_{m_{co}} - 10^{-14}$

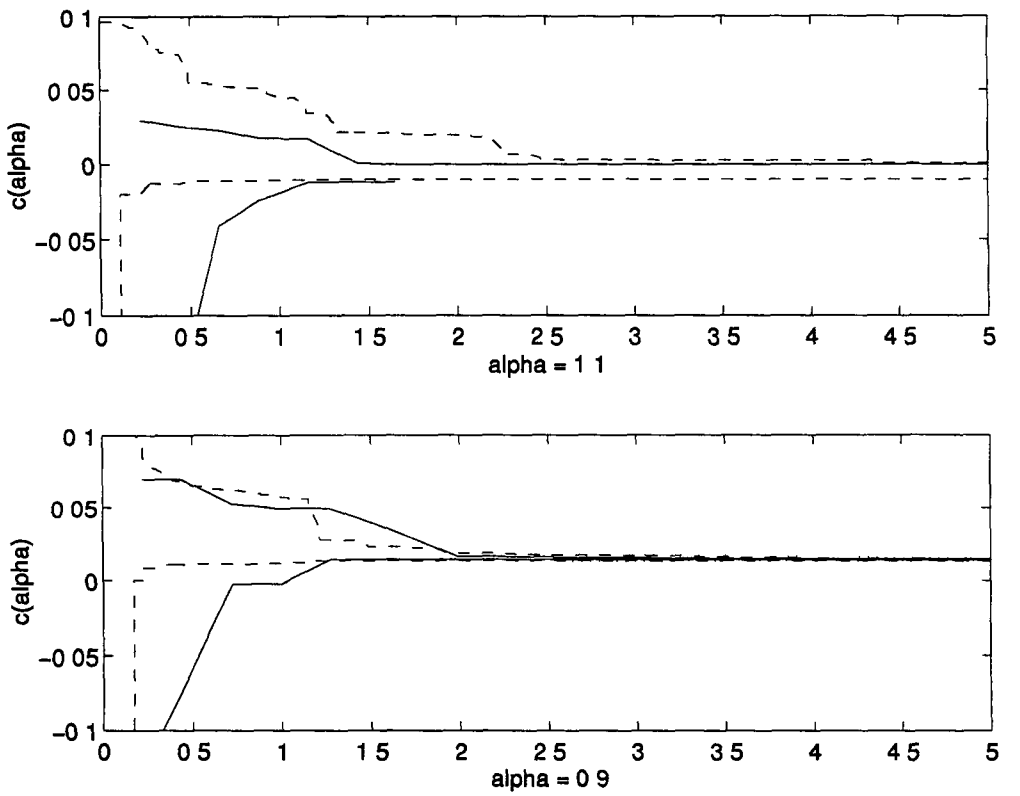


Figure 4 10 Bounds on $c(\alpha)$ using 1-norm dual (Full), 2-norm primal (Dashed) and 2-norm dual (Dot) methods

4.3.2 Relaxing Accuracy Requirements

Calculation of $k_{m_{co}}$ to the kind of accuracy levels detailed above is sometimes unnecessary Given that there is less software overhead involved in both 2-norm methods, either may give quicker answers to "rough" proximity problems Consider a typical example of a matrix with $k_{m_{co}} = 1$ generated using Fan's construction Two candidate α's were used, $\alpha_a = 9$ and $\alpha_b = 1\ 1$ The bottom graph of Fig 4 10 shows how, for α_a, the 2-norm primal algorithm is the quickest to determine that $0 \notin Co(W(\alpha_a))$ Clearly, at this level of accuracy, the time required for convergence is not a problem The top graph of Fig 4 10 shows how, at lower accuracy levels, the 2-norm dual approach can terminate without finding a positive minimum eigenvalue This termination indicates that one does not exist For both α_a and α_b the 2-norm dual approach is outperformed by the other two methods

It is now shown that these results are typical of what can be expected in general An

indication of the limits at which the 2-norm approaches can be effective is also given Two distinct questions are considered

1 For a fixed value of α, how do the three algorithms fare at reducing the upper bound on $c(\alpha)$ to below an abitrarily small ϵ Whether this ϵ is 'small enough" is a heuristic decision that can be open to question

2 For a fixed value of ϵ, how do the three algorithms fare at determining that $c(\alpha) \leq \epsilon$ This is equivalent to asking how the performance of the various algorithms deteriorate as $\|\alpha - k_{m_{co}}\| \to 0$

A batch of 20 3×3 matrices were generated randomly This was to counter the possibility that Fan's construction may impose a possible bias in favour of one or other algorithm For each matrix, $k_{m_{co}}$ was scaled equal to 1, correct to seven decimal places Throughout, entirely similar results to those outlined in Fig 4 10 were achieved for the batch of matrices Fig 4 11 answers question 1 by illustrating the performance of the algorithms with a seed α that is 0 1% larger than $k_{m_{co}}$ The average time required for $c(\alpha) \leq \epsilon$ where

$$\epsilon = (\bar{\sigma}(M))^{-1} * 10^{\frac{-Index}{10}} * 10^{-5}$$

is graphed for each algorithm In the figure the full line represents the 1-norm dual method The dashed line represents the 2-norm primal and the dotted line represents the 2-norm dual *Index* corresponds to the x-axis in the figure The scaling procedure is necessary to yield a normalised measure of $c(\alpha)$ for a matrix M The nature of the 1-norm dual curve is indicative of a "jump" to zero, similar to that which occurs in Table 4 10 This verification that a non-zero hyperplane which would separate 0 from the $Co(W)$ does not exist, is typical of standard LP behaviour

The second question gives a better feel for when the 1-norm dual superiority really begins to be felt The 20 3×3 matrices used in the previous experiment were considered again Five different possible upper bounds were selected [1 1, 1 01, 1 001, 1 0001, 1 00001] All these values are greater than $k_{m_{co}}$ to varying degrees In each case, the time taken to verify that these were indeed valid upper and lower bounds on $k_{m_{co}}$ for each problem was recorded For this experiment the performance of the 1-norm dual algorithm is clearly superior, on average, when validating an upper bound on $k_{m_{co}}$ This superiority becomes more marked as the distance between $k_{m_{co}}$ and the

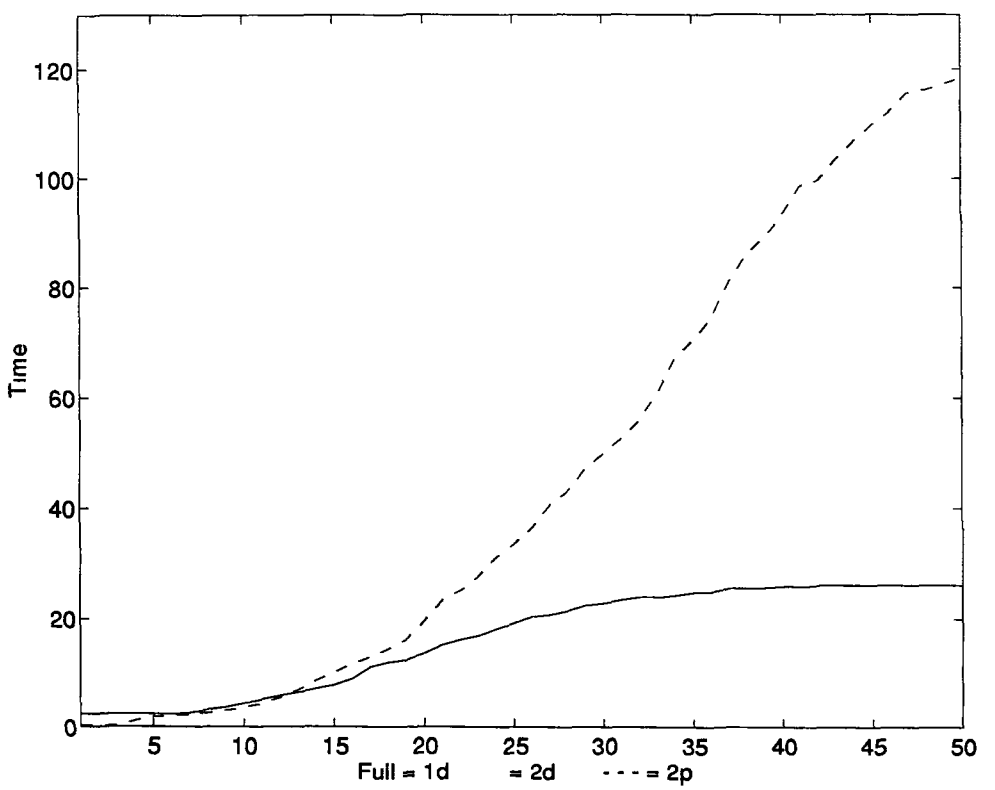


Figure 4.11 Time required for bounds on $c(\alpha) \leq \epsilon$ where $\alpha = 1.01 * k_{m_{co}}$

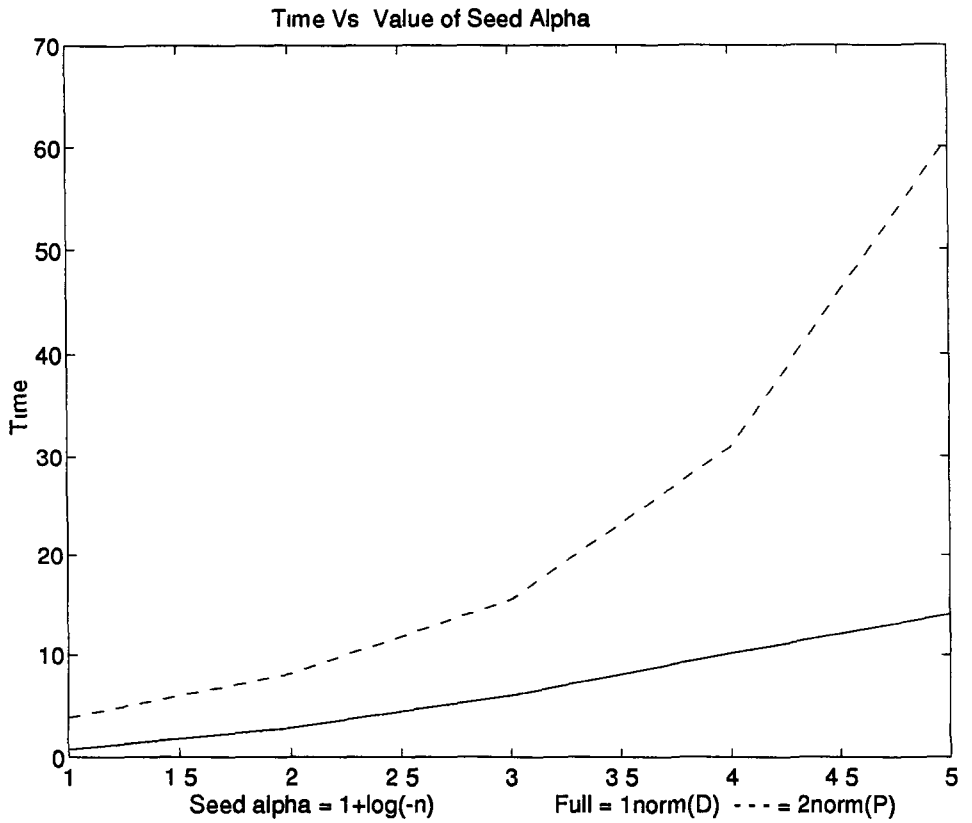


Figure 4 12 Time required for bounds on $c(\alpha) \leq \epsilon$ where α varies from 0 1% to 0 00001% $> k_{m_{co}}$

upper bound gets smaller Fig 4 12 illustrates that the time required by the 2-norm primal algorithm to reduce $c_2(\alpha)$ to less than an arbitrary ϵ increases exponentially as $\|\alpha - k_{m_{co}}\| \to 0$ In contrast the time required by the 1-norm dual algorithm shows a rate of increase that is close to linear

This experiment is repeated for similar batches of $n \times n$ matrices where $n \in [3, \quad, 8]$ Fig 4 13 demonstrates that these trends are repeated for increasing n where $n \leq 7$ At this point linear programming difficulties begin to impact unfavourably on the dual approach

It is also possible to demonstrate that the 1-norm dual approach catches up to its 2-norm primal counterpart when searching for positive minimum eigenvalues at higher accuracy levels To do this, the 20 3 $\times$ 3 problems were considered again Four different lower bounds were selected [0.9, 0.99, 0 999, 0 9999] For this experiment the 1-norm dual approach is outperformed by the 2-norm based primal algorithm for

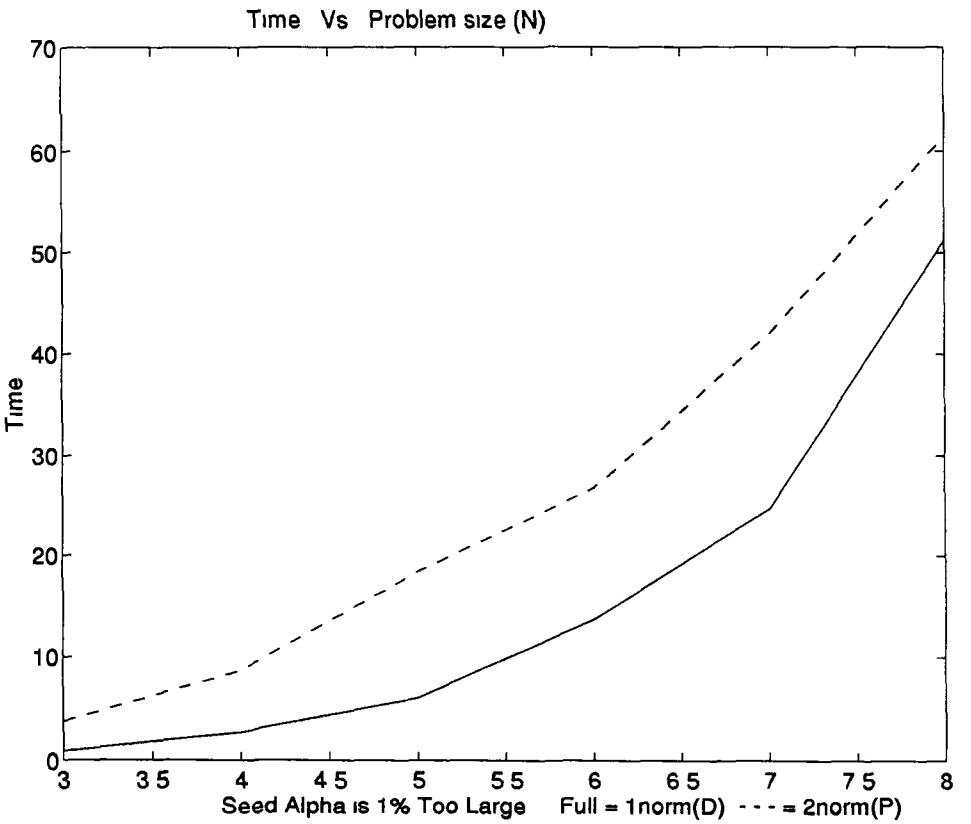


Figure 4 13 Time required for bounds on $c(\alpha) \leq \epsilon$ where $N \in [3, \quad, 8]$

these random problems However, there is a significant closing of the gap between the two approaches as α increases For $\alpha = 0\ 9999$, the difference in performance is negligible In all cases the 2-norm dual approach was outperformed by its 1-norm dual analogue on average A good starting vector is essential for a 2-norm dual approach Devoting time to locating such starting vectors may close this gap in performance However, it is always likely that problems will arise with certain matrices

4.3.3 Computing Bounds on μ_{co}

A level playing pitch comparison of the three different approaches was performed by determining upper and lower bounds on μ_{co} for a selection of random problems using a 386 PC Complex non-repeated scalar uncertainty was used throughout No pre-scaling of matrices was performed before analysis with each method Table 4 12 shows the amount of time required by the respective algorithms to obtain bounds that are within 1% of each other

An ∞ in a column indicates that the algorithm in question did not produce bounds within 1% of each other in the maximum allowed time The table lists times for 1-norm dual, 2-norm primal and dual approaches Also listed is a column which combines a 2-norm primal and dual approach This is necessary for convergence to be guaranteed in a 2-norm setting This column is calculated by doubling the computing time required by the algorithm that first arrives at a solution for a given value of α It is of interest to note that all the algorithms produced mutually consistent bounds for a given problem The tables indicate that the 1-norm dual algorithm is superior to 2-norm type algorithms when calculating upper and lower bounds on μ_{co}

Table 4.12 Comparison of Various Algorithms for the Determination of 1% bounds on μ_{co}.

Problem	1-norm Dual	2-norm Primal	2-norm Dual	Combined 2-norm
1	4 00400	∞	22 08000	40 10400
2	4 74000	∞	24 55700	46 95200
3	6 35500	∞	10 11800	17 79800
4	7 54700	∞	9 94700	18 67400
5	10 06200	∞	∞	∞
6	7 25000	∞	4 36100	5 44800
7	4 25700	∞	10 96300	21 92400
8	17 58700	∞	23 43600	5 43800
9	5 92700	∞	∞	∞
10	8 26000	∞	38 13500	9 73200
11	6 38800	∞	24 55700	4 51600
12	4 66400	∞	25 64500	43 72000
13	6 30500	∞	10 72100	18 17000
14	7 96400	∞	5 47600	7 24000
15	5 46500	∞	18 02100	19 50000
16	6 67900	∞	24 15600	21 98200
17	9 09500	∞	10 65000	9 06000
18	7 15100	∞	∞	∞
19	7 97500	∞	35 08000	70 16000
20	7 48600	∞	12 95700	25 90200

4.4 The μ, μ_{co} Gap

Throughout this thesis the emphasis has been on the computation of μ_{co} rather than μ To examine the nature of this gap in a practical situation, consider the problem introduced by De Gaston and Safonov [De Gaston 88] where μ was calculated exactly For the same problem bounds on μ_{co} were calculated using the new algorithm Fig 4 14 illustrates that the μ, μ_{co} gap can be significant This is particularly true

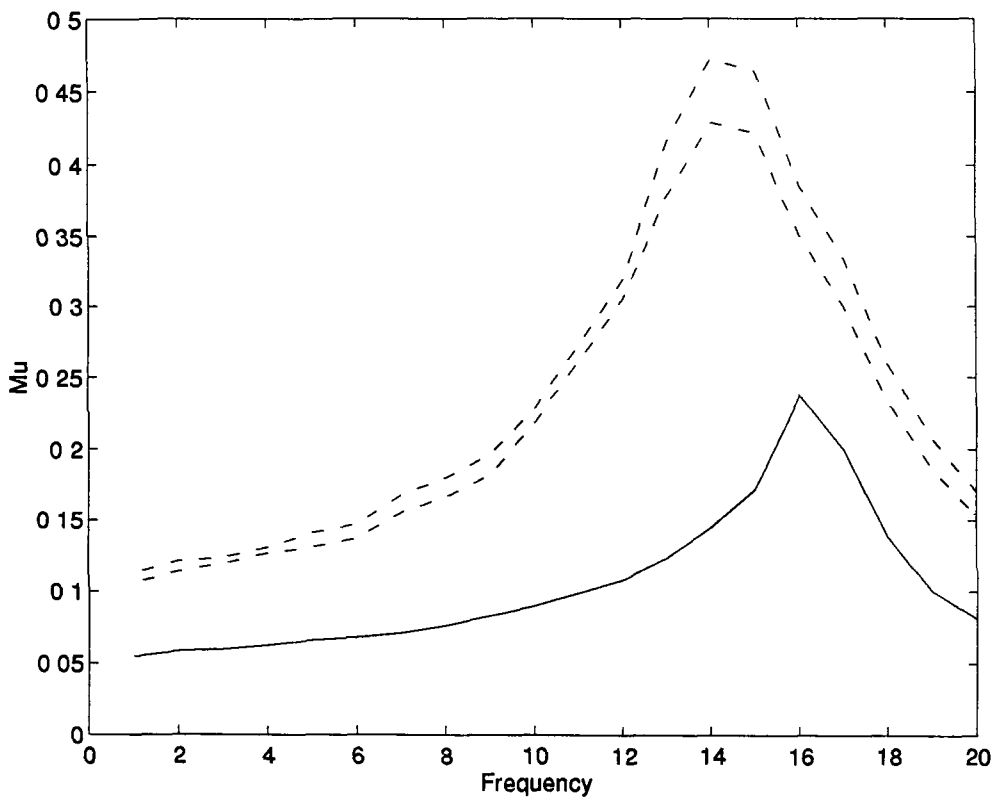


Figure 4 14 A practical problem which illustrates the gap between μ (Full) and μ_{co} (Dashed)

at the frequency of interest, i e , where μ is a maximum This suggests that while μ_{co} is indeed a reasonable bound on μ, work is necessary to reduce conservatism for practically motivated problems

4.5 Linear Programming Variations

Many different ways in which the significant percentage of time that is spent inside the linear program solver can be reduced have been considered in the previous chapter This section briefly mentions some of the computational experience gathered during the course of the project The treatment is no by means exhaustive but does suggest that while improvements to the computing times are possible, more work is required in this vast area before the LP solver section of the work can be regarded as complete

4.5.1 Interior Point Methods

Public domain Interior Point LP code due to Boyd *et al* [Boyd 94] was compared with the Numerical Recipes simplex code, implemented in *MATLAB*, on a UNIX platform Recent survey literature [Beran 95] suggests that Boyd's code is the best interior point solver available at the present time 20 3×3, 5×5 and 8×8 randomly generated problems were considered The flops required to obtain bounds within 1% of each other were recorded and are presented in Table 4 13

Table 4.13 Comparison of Simplex and Interior Point LP Solvers.

	Simplex		Interior Point		Percentage
N	*Average*	*Worst*	*Average*	*Worst*	Improvement
3	6 0305e+05	6 4538e+06	5 3273e+05	2 4869e+06	13 2
5	7 7891e+05	1 5771e+07	8 9730e+05	5 3771e+06	15 2
8	1 5374e+07	4 612e+08	1 2344e+07	9 1613e+07	28 3

The table shows that the Interior Point method offers superior computing times to a simplex based LP solver on average In addition the interior point method does not exhibit the same dramatic increase in flops required when $n \geq 7$ However the rate of increase of flops required with problem size is still exponential The code also exhibits a tendency, (in tandem with the simplex LP code offered in the *MATLAB* optimisation Toolbox), to have termination difficulties when α tends toward $k_{m_{co}}$

The numerical recipes algorithm is clearly more reliable when the LP cost is very small Indeed, the code due to Boyd tends to crash when $\|k_{m_{co}} - \alpha\| \leq 10^{-6}$ More work is necessary in this area

Analysis of the Simplex algorithm performance shows that computing times suffered more from increasing the number of constraints than the number of variables This suggests that research into the formation of a suitable dual problem, which would transpose constraints and variables, may be rewarding

4.5.2 Changing the D_k range

Some brief comments are now made about the nature of the solution hyperplanes for the linear programming problems that have been solved throughout this work When dealing with complex only uncertainty the following optimisation problem is solved, subject to suitable constraints on the A_k matrices

$$\max_{d_{k\in[0,+1]}} \lambda_{min}(\sum_k d_k A_k) \tag{4 1}$$

Similarly for the case of mixed real/complex parameter uncertainty the problem at hand is to

$$\max_{d_{k\in[-1,+1]}} \lambda_{min}(\sum_k d_k A_k) \tag{4 2}$$

It should be noted that expr (4 2) can also solve the purely complex problem The Numerical Recipes LP solver includes the facility to deal with degeneracy As discussed earlier, lots of zeros in the problem tableau should be avoided as they tend to introduce numerical problems To this end, the following general linear programming problem was considered -

$$\max_{f_{k\in[0,2]}} \lambda_{min}(\sum_k f_k A_k \; - \; \sum A_k) \tag{4 3}$$

At the end of an inner loop iteration this means that the solution hyperplane should not contain a surfeit of zero elements A representative sample of work has been carried out solving the problems related to expressions (4 2) and (4 3) There seems to be no appreciable difference between the performance of the LP solver on either question This was true on the average and for any exceptional cases that were considered However, expr (4 3) was chosen as the question to be solved throughout this work for intuitive reasons

4.5.3 Settling for Sub-Optimal Feasible Vectors

Terminating the linear program early did in fact offer a marginal improvement on the basic algorithm outlined in the previous chapter. However, the level of improvement was considerably inferior to the other conflicting possible improvements (i.e., tight constraints, matrix pencil theory) for purely complex uncertainty. There was also little or no benefit to be seen in the mixed case when compared with the, of necessity alternative, approach using the singular value bound. Therefore, no comprehensive computational work investigating early linear program termination was carried out.

4.6 Summary

The accuracy and reliability of the new algorithm has been demonstrated on a large variety of random, pseudo-random and practically motivated complex and mixed uncertainty problems.

Analysis of computation times have shown that the improved form of the new algorithm is competitive, though slower, when compared with existing commercially available code for the determination of bounds on μ.

The new algorithm is more reliable than the MFD Toolbox code when convergence of the bounds to a certain level of accuracy needs to be guaranteed. The new algorithm also offers improvements, when compared with the μTools code, for the calculation of μ_{co}. This is particularly true with mixed uncertainty structures. The convergence properties of the new algorithm guarantee that bounds on μ_{co} can be specified within a user defined distance of each other. The new algorithm generates these bounds without recourse to a power algorithm. The new code also offers greater flexibility in that it allows a performance parameter to hit only user specified blocks of a problem matrix.

Linear programming difficulties begin to adversely affect the new algorithm's performance when problem size is greater than 7. The use of an interior point linear program solver can improve but not obviate these difficulties. These difficulties seem to be dependent more on the number of variables than the number of constraints which exist

in a given problem This suggests that the use of dual linear programming methods could be a useful direction for further improvement

A general proximity problem is motivated by the application of the Hahn-Banach Theorem to the question of stability analysis A 1-norm dual approach has been shown to be, in general, superior to its 2-norm based primal and dual counterparts when attempting to solve this problem Moreover, it has also been demonstrated that bounds which are accurate correct to the operating precision of the given computer platform are possible using a 1-norm dual approach

Chapter 5

Analysis of Filter Performance

No manufacturing process is perfect It is not possible to manufacture, for instance, capacitors with the *exact* desired capacitance The cost of circuit elements also increases rapidly with lower tolerance ratings This raises the question of how to ascertain what effect a 1% variation, say, in the value of these capacitors will have on the transfer functions of filters constructed using these elements Indeed in many engineering applications it may be wise to determine the worst case system response For example, a worst case upper and lower bound on filter gain in response to any possible (bounded) variation in component values allows *guaranteed* achievement of certain design specifications It is also desirable to have a precise idea of what can happen at each frequency of interest

The structured singular value μ, or more precisely its convex estimate μ_{co}, can be used to solve this problem In this chapter it is shown how to recast various filter sensitivity problems into equivalent system stability questions The framework required to automate the process for any linear system is discussed and typical examples using Butterworth and Chebychev filters are presented

5.1 Problem Formulation

This section explains how the problem is arranged so that μ-theory can be brought to bear on it Initially the simple low-pass RC filter of Fig 5 1 is used as an example For higher order filters it is convenient to introduce system building blocks so as to prevent repetition From a μ perspective however, no new point of principle arises for any linear filter whatever the complexity The RC filter thus conveys all the key ideas without the introduction of a huge amount of new nomenclature Hence, a discussion of the mechanics of these building blocks is postponed until later

5.1.1 Uncertainty Description

Consider a system that is finite dimensional, linear, time invariant (FDLTI), and with n uncertain parameters $\Delta_1, \quad, \Delta_n$ embedded within it Although uncertain, these parameters are constrained to lie within a certain set Let $\mathcal{D}$ denote this set, the set of perturbations to be considered The set $\mathcal{D}$ is, of course, problem specific In this case, each $\Delta_1, \quad, \Delta_n$ can be thought of as a real perturbation to the ideal parameter values, and will be viewed as an n-tuple $\Delta = \mathrm{diag}(\Delta_1, \quad, \Delta_n)$ It will prove convenient to view the n-tuple as being arranged as a diagonal matrix Δ The system obtained when each $\Delta_i = 0$ is termed the **nominal** or the **unperturbed system** For the simple RC filter this unperturbed system would be the familiar

$$\frac{b(s)}{a(s)} = \frac{1}{1+RCs}$$

By convention, and without loss of generality, the bounds on each parameter may be arranged to be unity So each Δ_i can take on any value in the interval $[-1, +1]$ This leaves a family of systems, one system for every permissible perturbation $\Delta \in \mathcal{D}$ applied to the nominal system Suppose that the nominal or specified values of the resistor and capacitor are R_0 and C_0, and the actual values are R and C Suppose further that both elements have a tolerance rating of $\pm$ 10% A Δ_i is associated with each uncertain component Since each $\Delta_1, \quad, \Delta_n$ is constrained to lie within the set $[-1, +1]$, an appropriate scaling is required to reflect the tolerances for each component. The actual component values can then be expressed as

$$R = R_0(1 + 0\ 1\Delta_1), \qquad C = C_0(1 + 0\ 1\Delta_2)$$

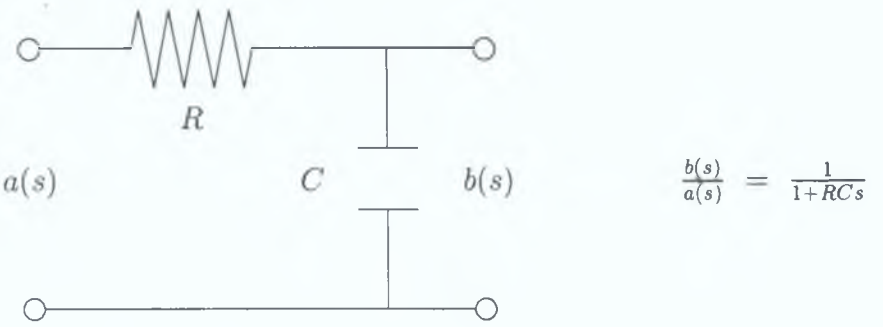


Figure 5.1: Low pass RC Filter with elements that may vary, at worst, by 10%.

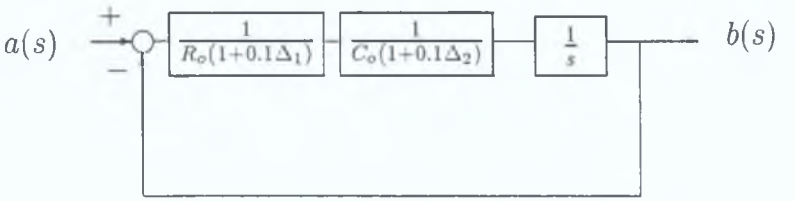


Figure 5.2: Equivalent block diagram representation of Fig. 5.1.

(where the 0.1 in this example represents a 10% tolerance on each component). Both parameters Δ_1 and Δ_2 have values which are unknown, but which are known to lie in the interval $[-1, +1]$. Thus, the 2×2 diagonal matrix Δ lies in the set

$$\mathcal{D}_{eg} = \{\Delta \in \mathcal{C}^{2\times 2} | \Delta = \text{diag}\{\Delta_1, \Delta_2\}, \Delta_1, \Delta_2 \in [-1, +1]\}$$

The transfer function of the filter is

$$\frac{b(s)}{a(s)} = \frac{1}{1 + RCs} = \frac{1}{1 + R_0C_0(1 + 0.1\Delta_1)(1 + 0.1\Delta_2)s} = G(s, \Delta)$$

The ideal or nominal transfer function can be denoted by $G(s, 0)$. This equation describes a FDLTI system with 2 uncertain parameters embedded in it. As Δ ranges over $\mathcal{D}_{eg}$, $G(s, \Delta)$ traces out a family of systems. A block diagram representation of $G(s, \Delta)$ is shown in Fig. 5.2.

5.1.2 The Diagonal Perturbation Formulation

It is possible to "extract" the uncertainty that is embedded within Fig. 5.2. This extracted uncertainty is then viewed as an external Δ acting on the nominal system, i.e., the nominal system is what is left behind when this uncertain Δ has been extracted. To illustrate, look at the equivalent realisations of the system of Fig. 5.2

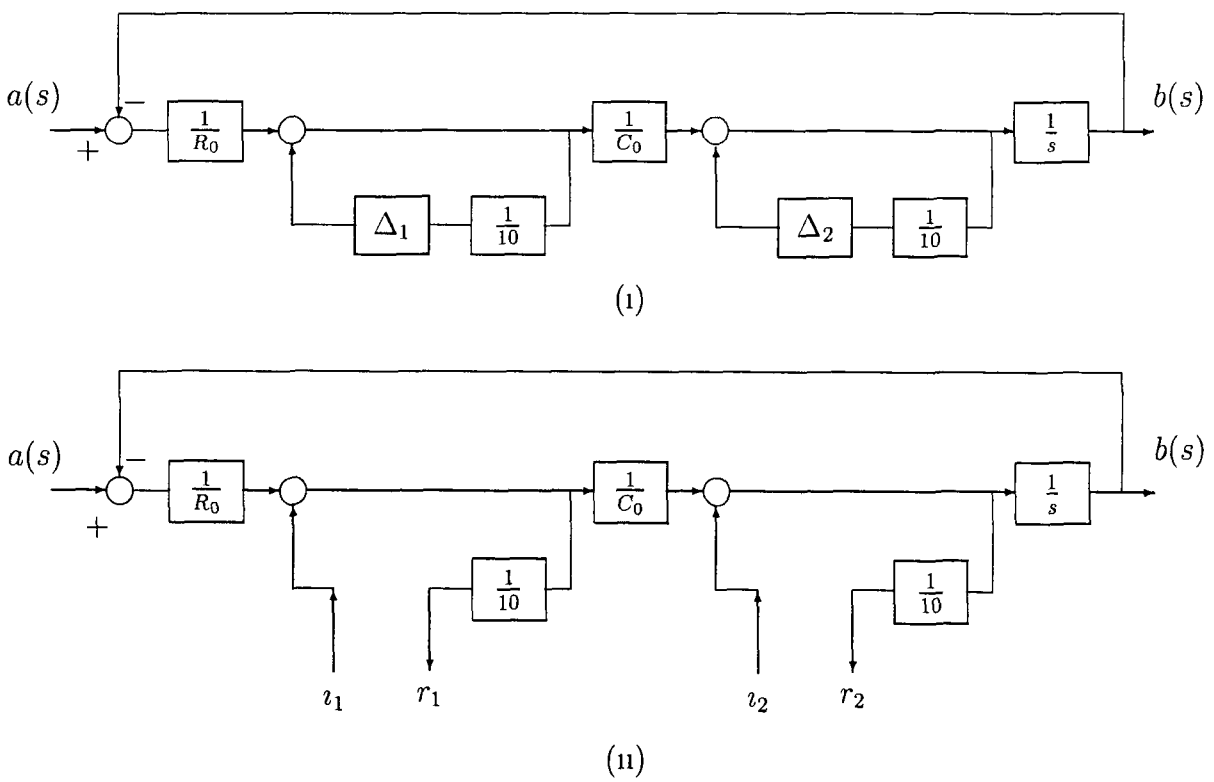


Figure 5 3 Extraction of uncertain Δ's from simple RC filter of Fig 5 2

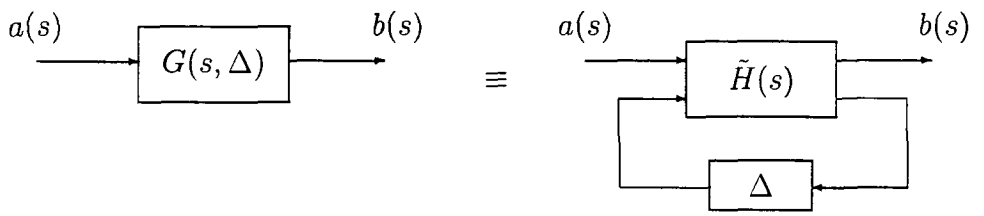


Figure 5 4 The Diagonal Perturbation Formulation (DPF)

which are shown in Fig 5 3 Each Δ_i is extracted from the system as in Fig 5 3(ii) The output from Δ_i will form the external input i_i to an augmented nominal system Similarly the corresponding output r_i from this nominal system is the input to Δ_i By calculating the MIMO transfer function from $a(s), i_1$ and i_2 to $b(s), r_1$ and r_2 one can move to an equivalent representation of $G(s, \Delta)$ using two blocks and a feedback loop Note that the Δ is still $\in \mathcal{D}_{eg}$, but it is now *outside* the augmented form of $G(s, \Delta)$ This augmented nominal system is denoted as $\tilde{H}(s)$, as in Fig 5 4, to emphasise its independence from Δ The diagonal structure of Δ ensures that each Δ_i is associated with one uncertain element only This rearranged representation of the original system $G(s, \Delta)$ is called its *Diagonal Perturbation Formulation (DPF)* Any linear transfer function can be expressed in terms of its DPF [El Ghaoui 91]

For the simple RC filter, this process yields

$$\Delta = \begin{pmatrix} \Delta_1 & 0 \\ 0 & \Delta_2 \end{pmatrix}, \qquad \tilde{H}(s) = \begin{pmatrix} 1 & -0\,1R_0C_0 & -0\,1R_0 \\ s & -0\,1R_0C_0s & -0\,1R_0s \\ C_0s & -0\,1C_0 & -0\,1R_0C_0s \end{pmatrix} \frac{1}{1 + R_0C_0s}$$

It can be already seen how rearranging a problem into its DPF can require a good deal of very tedious work

5.1.3 Formal Statement of the Filter Robustness Problem

A precise definition of worst case filter performance is now given using the terminology that has been introduced thus far

The maximum transfer function gain from input $a(s)$ to output $b(s)$ may be written as

$$\max_{\Delta \in \mathcal{D}} \left| \frac{b(s)}{a(s)} \right| = \max_{\Delta \in \mathcal{D}} |G(s, \Delta)| = |G_{max}(s)| \qquad (5\,1)$$

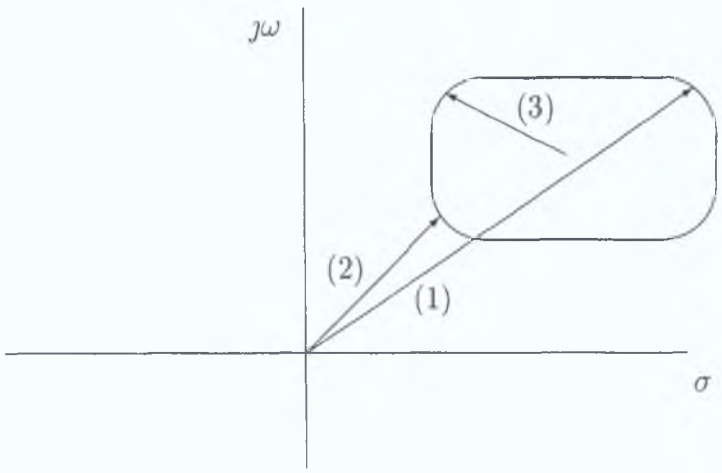


Figure 5.5: Illustration of the different perturbations that cause the worst possible deviation from the nominal response.

$|G_{max}(s)|$ thus determines the maximum filter gain over all frequency as Δ ranges over all permissible perturbations in $\mathcal{D}$.

Similarly, the minimum transfer function gain is given by

$$\min_{\Delta \in \mathcal{D}} |\frac{b(s)}{a(s)}| \;=\; \min_{\Delta \in \mathcal{D}} \; |G(s,\Delta)| \;=\; |G_{min}(s)| \tag{5.2}$$

which will determine the minimum possible filter response for all values of frequency. Solution of the optimisation problems in eqns. (5.1) and (5.2) will determine two distinct Δ's that bound worst case filter gain above and below.

It is also possible to determine the Δ that will yield the maximum deviation, in a Euclidean distance sense, from the nominal response on a polar plot.

$$\max_{\Delta \in \mathcal{D}} \; |G(s,\Delta) - G(s,0)| \;=\; |G_{dev}(s)| \tag{5.3}$$

Therefore $|G_{dev}(s)|$ corresponds to the maximum difference between the nominal and the actual filter transfer function on a polar plot. To illustrate how the solution of eqn. (5.3) does not necessarily coincide with one of the perturbations in eqns. (5.1) and (5.2) consider Fig. 5.5. The figure shows the nominal response of the filter forming the centre of a set in the complex plane. It should be noted that this set is a considerable simplification of the uncertainty that exists in this problem. Notwithstanding this, the figure serves to illustrate the point at hand. The maximum and minimum gains at this frequency will correspond to the maximum (1) and minimum (2) distances from

the origin to the set The maximum deviation from the nominal, (3), corresponds to the maximum Euclidean distance from the centre of the set to a boundary point on a polar plot Thus, distance (3) can be seen as the farthest point from the nominal that a valid Δ can perturb the filter response to in gain *and* phase terms This consideration of phase information is the principal reason why it maybe useful to consider bounds other than those which arise from eqns (5 1) and (5 2)

5.2 An Equivalent Stability Problem

An analogy between the problem at hand and that of determining the effect of perturbations to a plant in feedback control systems is now drawn Any perturbation or uncertainty in the plant model has an effect on the filter transfer function from $a(s)$ to $b(s)$ This transfer function can be viewed as the closed loop response of $\tilde{H}(s)$, (the "open loop" response) which is connected via a perturbation Δ on a feedback loop Viewing the uncertain elements as "external" in a feedback system suggests an analogy with μ Indeed, μ-analysis can exploit the *a priori* knowledge that exists about the internal structure of the uncertainty in a system and treat it in a worst case sense It will now be shown how, when appropriate constructions are used, μ is the non-conservative measure of the worst case effect of uncertainty on filter behaviour

5.2.1 Application of the Robust Performance Theorem

The filter sensitivity problem is now recast as an equivalent robust stability question The argument to be used is based on an application of the **Robust Performance Theorem** [Stein 82], which is now discussed briefly Consider the system of Fig 5 6 There are n uncertain parameters $\Delta_1, \quad , \Delta_n$ which correspond to variations in component values These Δ's are constrained to be real valued The additional "fictitious" uncertain parameter in this case will be a bound on the gain of the filter, $k\Delta_f$ say, as illustrated in Fig 5 6 Here, k is a positive real scalar, and Δ_f is viewed as unknown, but is constrained to have modulus ≤ 1 at each frequency, i e ,

$$\Delta_f \in \mathcal{D}_f \quad \text{where} \quad \mathcal{D}_f = \{\Delta_f \in \mathcal{C} \mid , \ |\Delta_f(j\omega)| \leq 1\}$$

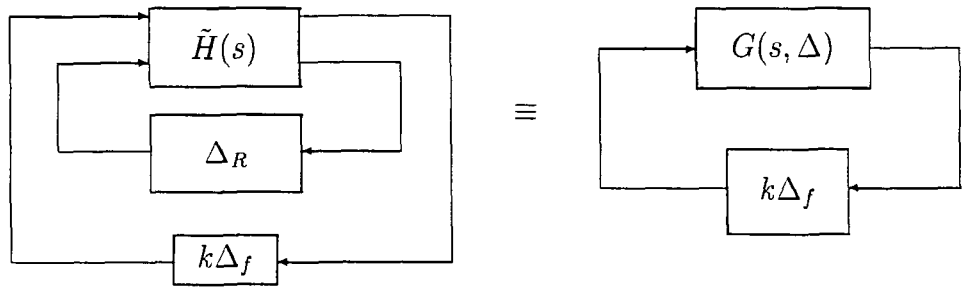


Figure 5.6 Incorporating fictitious performance parameter into the DPF

Unlike $\Delta_1, \quad , \Delta_n$, the fictitious parameter Δ_f is allowed to be complex valued The fictitious term $k\Delta_f$ is included as an additional element in the diagonal matrix $\tilde{\Delta}$ Therefore,

$$\tilde{\Delta} = \begin{pmatrix} k\Delta_f & 0 \\ 0 & \Delta \end{pmatrix}$$

$$\tilde{\mathcal{D}} = \{\tilde{\Delta} \in \mathcal{C}^{(n+1)\times(n+1)} | \tilde{\Delta} = \begin{pmatrix} k\Delta_f & 0 \\ 0 & \Delta \end{pmatrix}, \Delta \in \mathcal{D}, \Delta_f \in \mathcal{D}_f\}$$

Now there is an equivalent representation of $\tilde{H}(s)$ whose input/output pairs are connected exclusively by one $\tilde{\Delta}$ block Consider this self contained two block feedback loop Let the filter gain k be fixed and given for the moment By considering the Nyquist stability criterion, it is clear that if there is a $\Delta \in \mathcal{D}$ for which $|G(s,\Delta)| \geq k^{-1}$, then there is a $\Delta_f \in \mathcal{D}_f$ for which the system is unstable (having a loop gain ≥ 1) Conversely, if $|G(s,\Delta)| < k^{-1}$, for all $\Delta \in \mathcal{D}$, then the system is stable for all $\Delta_f \in \mathcal{D}_f$ (having a loop gain < 1 for every permissible perturbation) Thus, the maximum possible "size" of $|G(s,\Delta)|$ is bounded by k^{-1} if and only if a certain system is robustly stable As k is increased, the first value of k for which this feedback system may become unstable corresponds to the largest possible $|G(s,\Delta)|$ being k^{-1} There will therefore be a distinct value for k where instability occurs at each frequency This is an important feature in that one frequency is decoupled from another Information of this kind means that the engineer can now think in frequency response terms, an obvious benefit in the present context of filter design

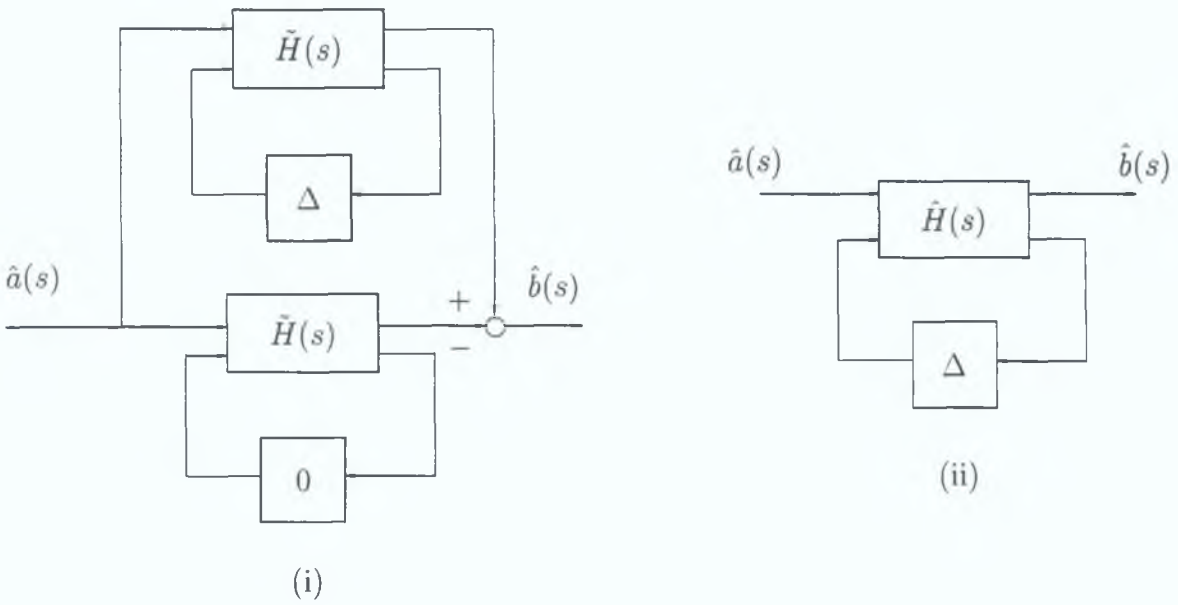


Figure 5.7: Determination of worst case deviation from nominal filter performance.

5.3 Computing Worst Case Filter Sensitivity

At this point, the problem at hand is to compute $G_{max}(s)$, $G_{min}(s)$ and $G_{dev}(s)$ over a frequency range of interest. Each of the above functions reduce directly to an evaluation of μ for a certain matrix.

Maximum Filter Gain, $G_{max}(s)$

Maximum filter gain can be computed from

$$G_{max}(s) \;=\; (\min_{\check{\Delta}\in\check{\mathcal{D}}}\{k|\,\det(1+\check{\Delta}(s)\check{H}(s))=0\})^{-1} \tag{5.4}$$

which corresponds to the smallest k for which the system may be unstable (i.e., not robustly stable). Thus,

$$G_{max}(s) \;=\; \mu(\check{H}(s)) = k_m^{-1}(\check{H}(s))$$

$G_{max}(s)$ therefore yields the largest possible value of $H(s,\Delta)$ at each value of frequency.

Maximum Euclidean Deviation, $G_{dev}(s)$

The perturbation that will cause this worst case Euclidean deviation from nominal performance, can be discovered by consideration of the arrangement in Fig. 5.7(i). In the figure, using a DPF of $G(s,\Delta)$, the ideal filter response has been subtracted from the actual filter response. Clearly this gives a different transfer function, say from from $\hat{a}(s)$ to $\hat{b}(s)$, which will be of the form

$$G(s,\Delta) - G(s,0)$$

where Δ ranges over all Δ's in $\mathcal{D}$. Thus the problem at hand is to determine, at each frequency of interest

$$\max_{\Delta\in\mathcal{D}} |G(s,\Delta) - G(s,0)| \;=\; G_{dev}(s) \tag{5.5}$$

Fig. 5.7 illustrates equivalent representations of this problem. Standard algebraic manipulations can be used to move between Fig. 5.7(i) and Fig. 5.7(ii). Maximum filter deviation from nominal performance can be computed by determination of the smallest k that will make the loop in Fig. 5.7(ii) go unstable. Thus

$$G_{dev}(s) \;=\; (\min_{\tilde{\Delta}\in\tilde{\mathcal{D}}}\{k|\,\det(1+k\tilde{\Delta}(s)\hat{H}(s))=0\})^{-1} \tag{5.6}$$

Therefore,

$$G_{dev}(s) \;=\; \mu(\hat{H}(s)) = k_m^{-1}(\hat{H}(s))$$

Notice how the uncertainty set description is unchanged for either eqn. (5.4) or eqn. (5.6)

Minimum Filter Gain, $G_{min}(s)$

A problem with the application of the robust performance theorem arises when attempting to determine the minimum possible filter gain. The theorem is only of use in finding the maximum allowable gain before a loop goes unstable. In order to determine the minimum gain from $G(s,\Delta)$ it is necessary to place it on the feedback path of a suitable system such as the one illustrated in Fig. 5.8. Again each of the representations in the figure are equivalent. Note that the ϵ in this figure is a constant

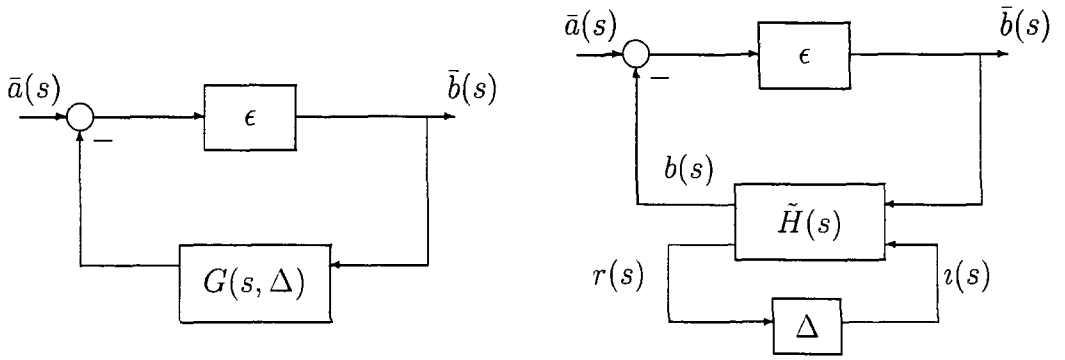


Figure 5 8 In order to determine the minimum gain from $G(s,\Delta)$ it is placed on the feedback path of a suitable closed loop system

gain term which can be assumed to be arbitrarily large Evaluating the closed loop response of Fig 5 8(i) yields

$$\frac{\bar{b}(s)}{\bar{a}(s)} = \frac{\epsilon}{1+G(s,\Delta)\epsilon}$$

Thus for large ϵ

$$\frac{\bar{b}(s)}{\bar{a}(s)} \approx \frac{1}{G(s,\Delta)}$$

Evaluation of this closed loop system shows that its gain will be a maximum when the gain of $G(s,\Delta)$ is a minimum It is instructive to consider the loop equations in Fig 5 8(ii) Clearly,

$$\bar{b}(s) = \epsilon(\bar{a}(s) - b(s))$$

$$\Rightarrow \bar{b}(s) = \epsilon(\bar{a}(s) - (\tilde{H}_{11}\bar{b}(s) + \tilde{H}_{12}\imath(s)))$$

It should also be noted that

$$r(s) = \tilde{H}_{21}\bar{b}(s) + \tilde{H}_{22}\imath(s)$$

Rearranging it can seen that

$$\bar{b}(s)(1+\epsilon\tilde{H}_{11}) = \epsilon(\bar{a}(s) - \tilde{H}_{12}\imath(s))$$

$$\Rightarrow \bar{b}(s) = \epsilon(1+\epsilon\tilde{H}_{11})^{-1}(\bar{a}(s) + \tilde{H}_{12}\imath(s))$$

$$\Rightarrow \bar{b}(s) = \epsilon(1+\epsilon\tilde{H}_{11})^{-1}\begin{bmatrix} 1 & -\tilde{H}_{12} \end{bmatrix}\begin{bmatrix} \bar{a}(s) \\ \imath(s) \end{bmatrix}$$

Similarly,

$$r(s) = \epsilon(1+\epsilon\tilde{H}_{11})^{-1}\left[\begin{array}{cc} \tilde{H}_{21} & -\tilde{H}_{21}\tilde{H}_{12}+\frac{1+\epsilon\tilde{H}_{11}}{\epsilon}\tilde{H}_{22}\end{array}\right]\left[\begin{array}{c}\bar{a}(s)\\ i(s)\end{array}\right]$$

Let the transfer function matrix between output vector $\bar{b}(s), r(s)$ and input vector $\bar{a}(s), i(s)$ be denoted as $\bar{H}(s)$ Thus,

$$\left[\begin{array}{c}\bar{b}(s)\\ r(s)\end{array}\right] = \left[\begin{array}{cc}\bar{H}_{11} & \bar{H}_{12}\\ \bar{H}_{21} & \bar{H}_{22}\end{array}\right]\left[\begin{array}{c}\bar{a}(s)\\ i(s)\end{array}\right]$$

where

$$\left[\begin{array}{cc}\bar{H}_{11} & \bar{H}_{12}\\ \bar{H}_{21} & \bar{H}_{22}\end{array}\right] = \epsilon(1+\epsilon\tilde{H}_{11})^{-1}\left[\begin{array}{cc}1 & -\tilde{H}_{12}\\ \tilde{H}_{21} & -\tilde{H}_{21}\tilde{H}_{12}+\frac{1+\epsilon\tilde{H}_{11}}{\epsilon}\tilde{H}_{22}\end{array}\right]$$

Since the system has now been rearranged into a a robust stability question of standard form, the Robust Performance Theorem can be applied Thus,

$$G_{min}(s) = (\min_{\tilde{\Delta}\in\tilde{\mathcal{D}}}\{k|\det(1+k\tilde{\Delta}(s)\bar{H}(s))=0\})^{-1} \qquad (5\ 7)$$

where such a k will correspond to the minimum gain possible from $G(s,\Delta)$ Therefore,

$$G_{min}(s) = \mu(\bar{H}(s)) = k_m^{-1}(\bar{H}(s))$$

5.4 The DPF for Any Ladder Filter

To be useful, this process needs to be expanded to circuits of a non-trivial nature This section considers the necessary extensions required to automate the generation of the DPF for an arbitrary ladder filter In any ladder realisation there are generally only a few basic building block elements to consider These blocks are connected in series to generate higher order filters For a Butterworth or Chebychev design there are only four to consider, namely

(i) The Source Resistance

(ii) The Series Inductor

(iii) The Parallel Capacitor

(iv) The Load Resistor.

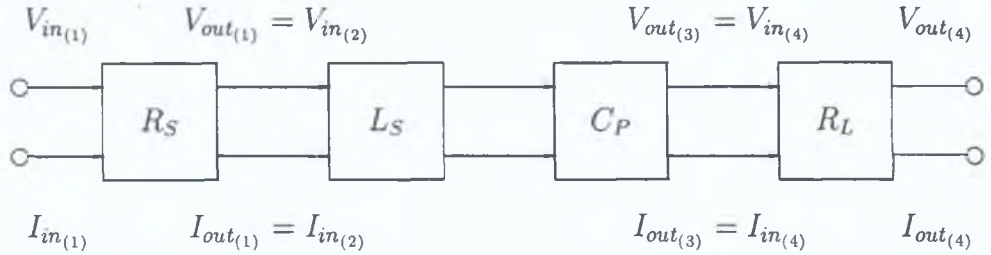


Figure 5.9: 2-port representation of the four building blocks used for a second order Butterworth design.

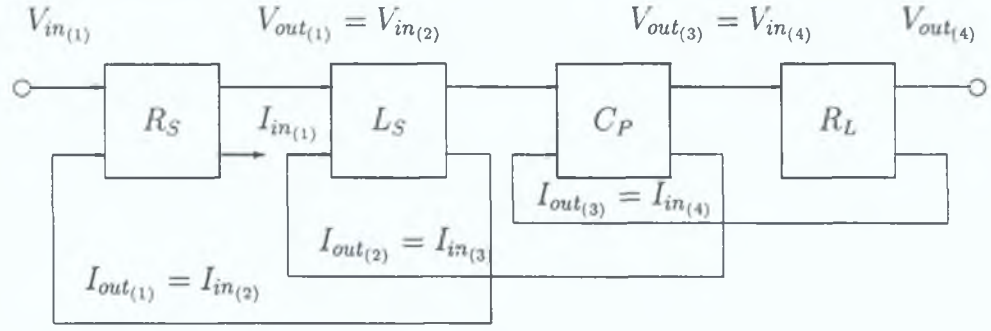


Figure 5.10: Fig. 5.9 redrawn with $I_{out_{(n)}}$ now acting as an input to a block.

Although totally different formulae are used to generate the component values in each case, only the four component blocks outlined above are used. Though the component blocks will be different, the same general principle also holds for other ladder filters. Each building block is represented in terms of the inverse of its h parameters using standard two port analysis techniques. Fig. 5.9 illustrates how these two port representations are connected together.

At this point a problem arises. The blocks in Fig. 5.9 contain two inputs V_{in} and I_{in} and two outputs V_{out} and I_{out}. The transfer function variation that is of interest is represented by the V_{in}, V_{out} pair. When an input voltage V_{in} is applied, the corresponding I_{in} is an unknown. Therefore the I_{in}, I_{out} pair represent variables that cannot be described *a priori*. This problem is solved by *reversing* the direction of the input/output pair using a feedback loop. This can always be done using simple algebraic manipulations. I_{out} is viewed as an input to the basic block and I_{in} as an output from the block. Feeding the output from block n back to the input of block $n-1$ results in the equivalent representation of Fig. 5.10. It can be seen from this figure that $I_{in_{(1)}}$ can now be regarded as a redundant output from the system.

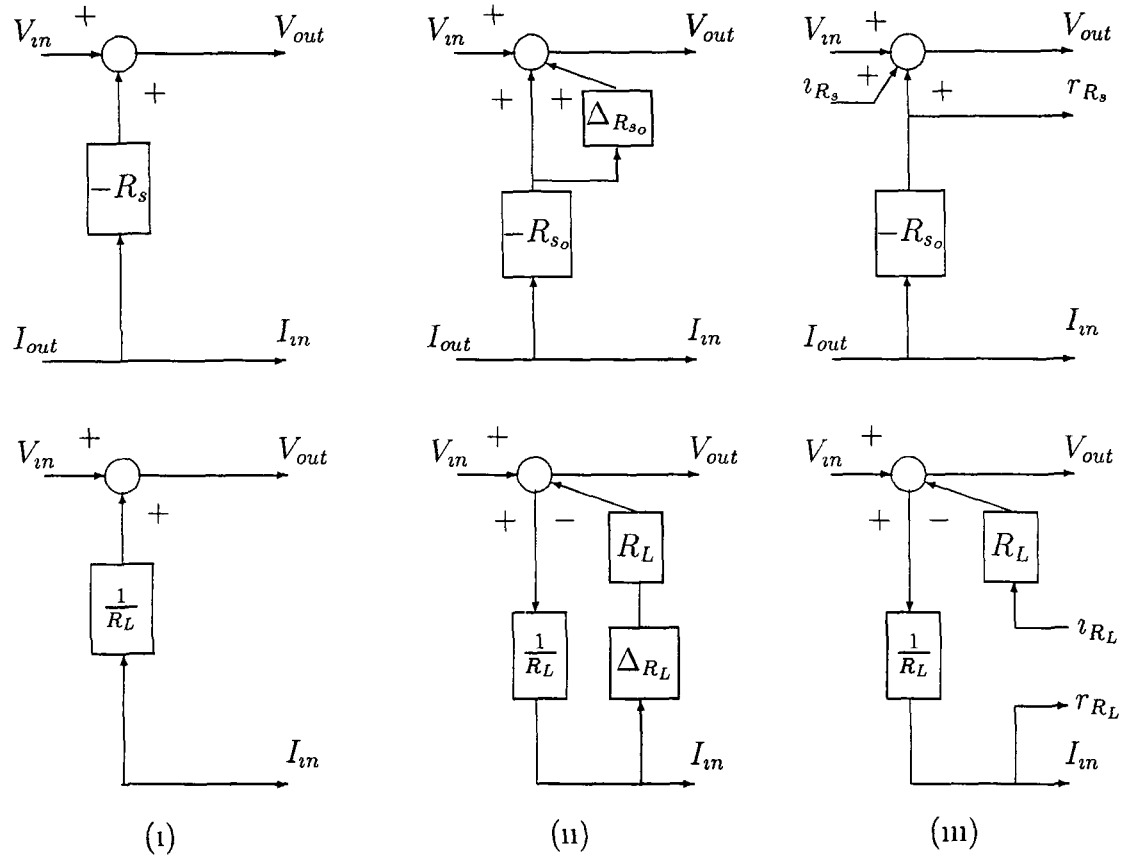


Figure 5 11 Steps to the DPF for the 2-port representation of source and load resistances

It is necessary to generate the equivalent DPF for each building block Figs 5 11 and 5 12 show how this is achieved in each case Again, one introduces uncertainty as in Fig 5 11 (Column (ii)), and then extracts it as in Fig 5 11 (Column (iii)) In this way the problem is reformulated as an augmented nominal system $\tilde{H}(s)$ that is "hit" by an external Δ Any Butterworth or Chebychev filter is made up of some form of interconnection of these elements For example the voltage output from the source resistor must form the voltage input to the first series inductor element In turn the output of this series inductor element is fed back to the input of the source resistor stage In the general case, this may be achieved using an *interconnection matrix* Fig 5 13 shows how the outputs from one element can become the inputs to another by using the interconnection matrix N N will contain only ones or zeros, with any row containing at most one non-zero element

Fig 5 13 can be represented equivalently by Fig 5 14 In Fig 5 14 one should view the

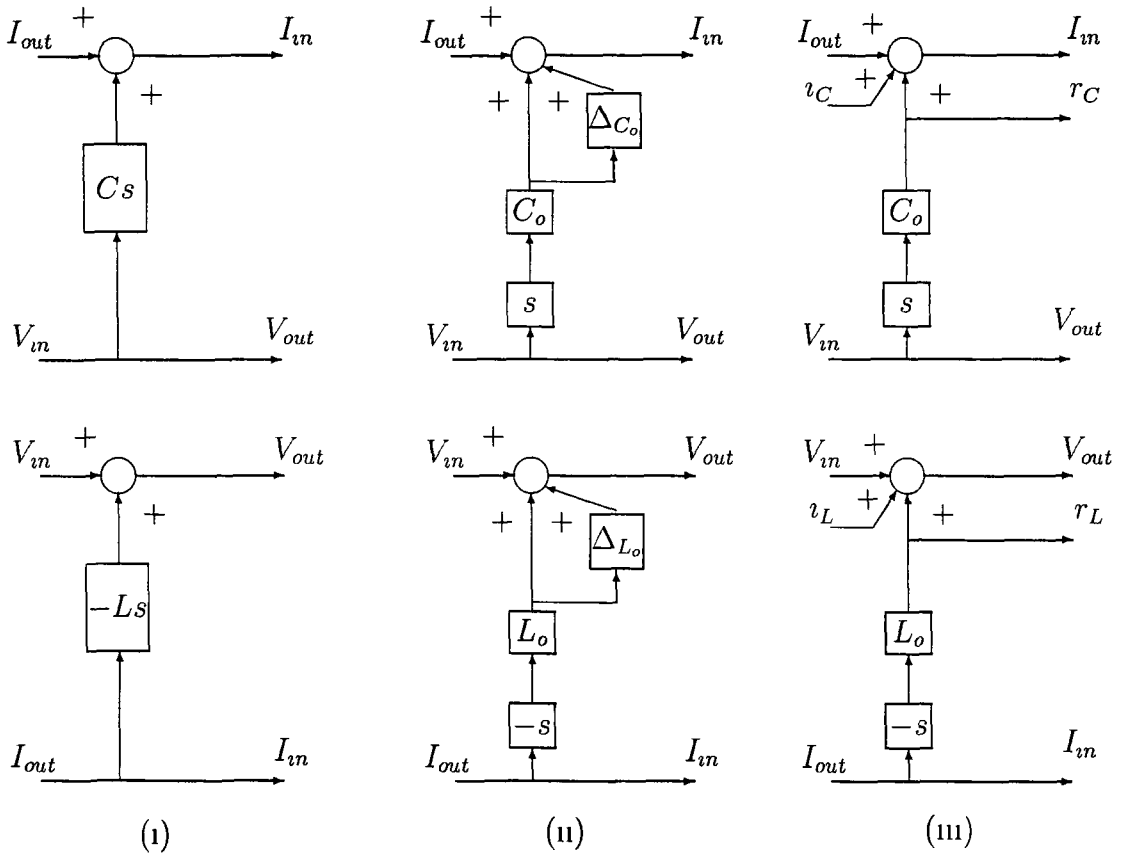


Figure 5 12 Steps to the DPF for the 2-port representation of series inductance and parallel capacitance

connection path between N and $\tilde{H}(s)$, the augmented "nominal" system, as a vector of dimension $2n$ Again, standard algebraic manipulation will convert the system of Fig 5 14 to that of Fig 5 4

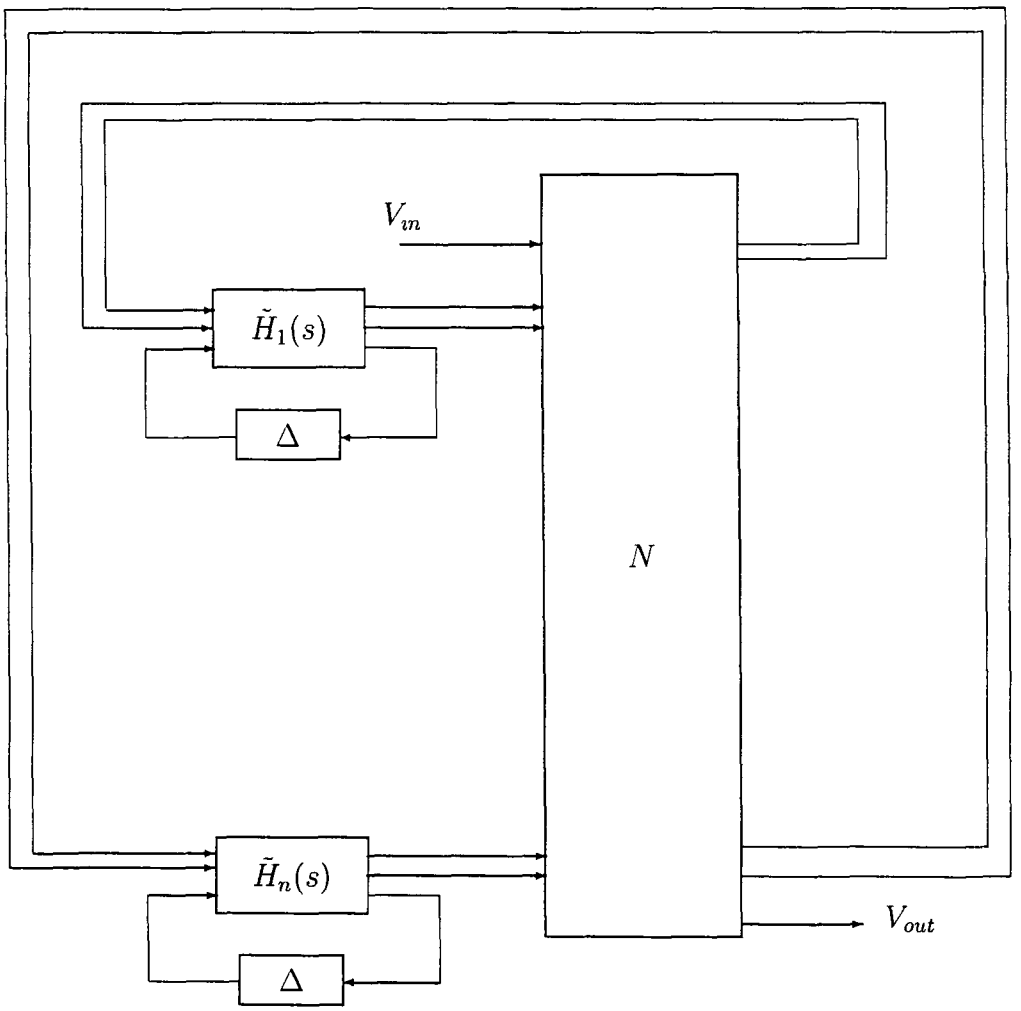


Figure 5 13 The use of an interconnection matrix for larger problems

5.5 Two Examples

The approach outlined in the preceding sections is now applied to the following two specific examples The design requirements are as follows -

Butterworth filter, order = 3
Chebychev filter, order = 3

Cutoff frequency = 1 kHz
Bandpass Ripple (Chebychev only) = 10%
Source Resistance = 10 Ω

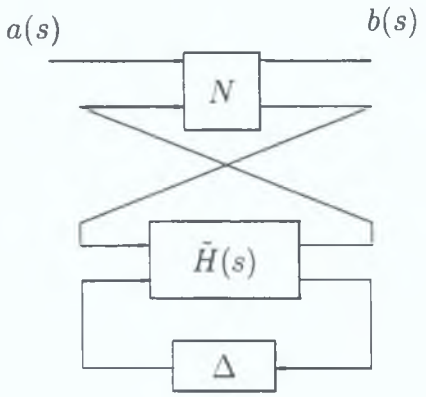


Figure 5.14: Equivalent representation of the system in Fig. 5.13.

Load Resistance = 10 Ω
Circuit element tolerances = 1%, 5% and 10%

Standard doubly terminated LC ladder realisations were used. Initially $G_{max}(s)$ and $G_{min}(s)$, the maximum and minimum possible filter gain respectively, are considered when the component values are allowed to vary. The code used provides upper and lower bounds on μ_{co}. Thus, the full lines either side of the nominal response in Figs. 5.15 and 5.16 represent an upper bound on $G_{max}(s)$ and a lower bound on $G_{min}(s)$.

$G_{dev}(s)$ is plotted for Butterworth and Chebychev in Figs. 5.17 and 5.18 respectively. In all cases these bounds are derived by estimating μ_{co} at each frequency. As a consequence the upper bound will err on the side of caution in terms of filter performance. This allows guarantees to be made about how well a certain filter will perform when component values vary. The true μ will lie somewhere between this bound and the lower bound generated by grid or random search techniques. To illustrate how large this gap can be consider Figs. 5.19 and 5.20. These show how a random search and μ_{co} yield different bounds on filter performance. By their nature, random or grid search methods will yield a result that under estimates worst case filter gain. A distinct advantage of μ-analysis is the way that it offers a repeatable and non-probabilistic *upper* bound on worst case filter sensitivity. The bounds generated using the μ-theory offer a considerable improvement on conventional bounds generated using differential sensitivity calculations. μ-analysis fully addresses the effects of multiple component uncertainties, including cross-coupling effects and inter-dependencies, an issue which differential sensitivity analysis avoids.

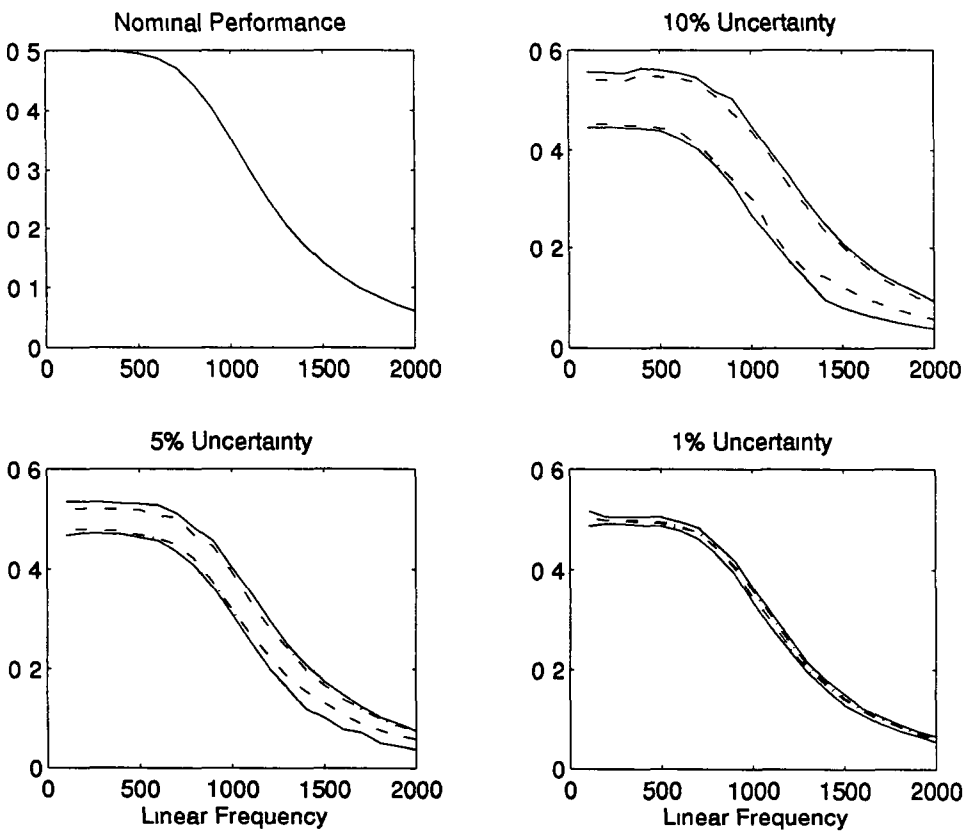


Figure 5 15 Upper(*Full*) and Lower (*Dash*) bounds on $G_{max}(s), G_{min}(s)$ for a Butterworth filter of Order 3

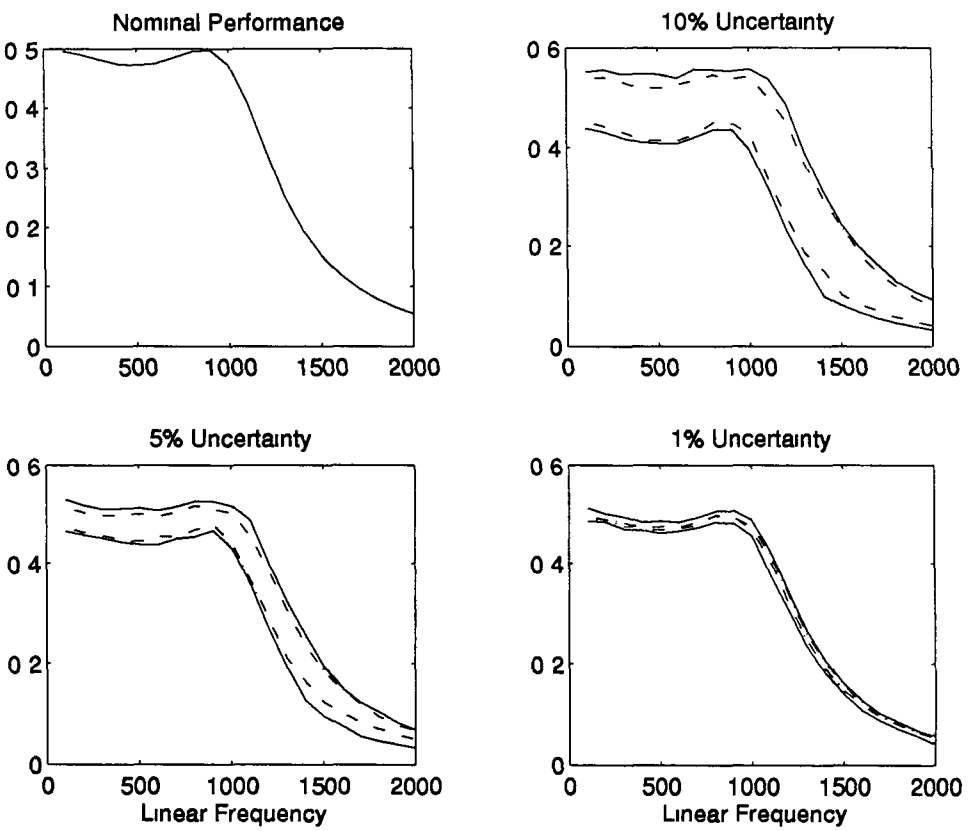


Figure 5 16 Upper(*Full*) and Lower (*Dash*) bounds on $G_{max}(s), G_{min}(s)$ for a Chebychev filter of Order 3

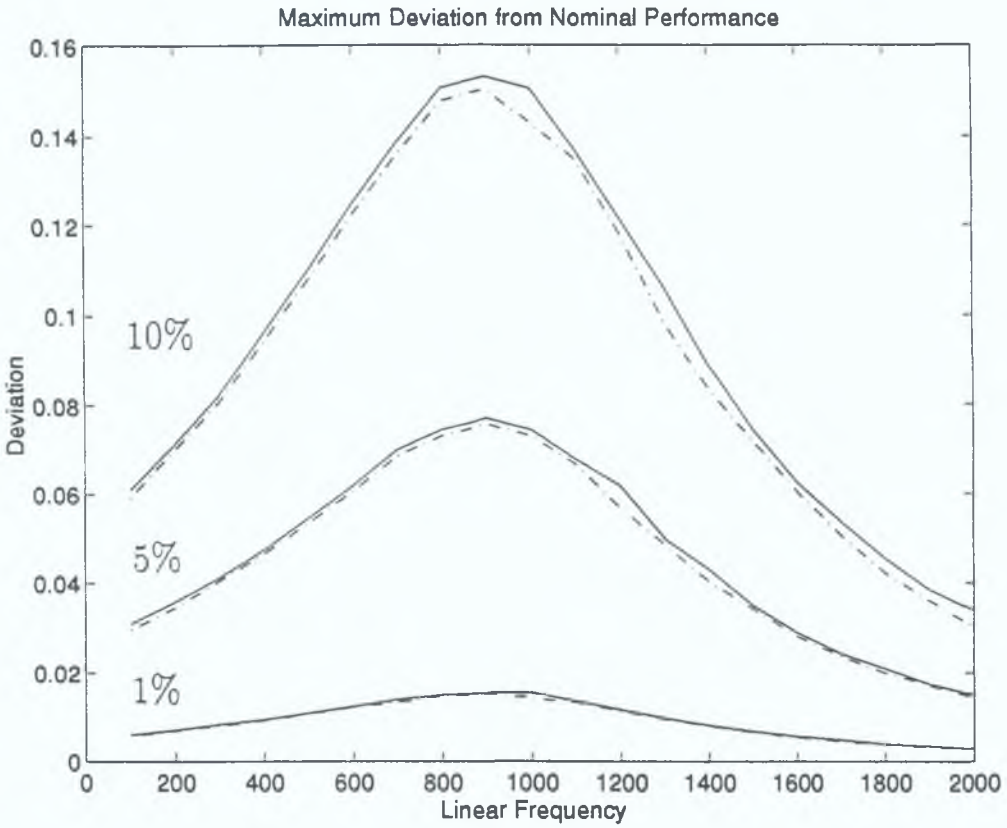


Figure 5.17: Upper(*Full*) and Lower (*Dash*) bounds on $G_{dev}(s)$ for a Butterworth filter of Order 3.

The graphs suggest that Chebychev filters are more sensitive to component variations than their Butterworth counterparts. The effect on the bandpass region is particularly noticeable, especially near the cutoff frequency. The uncertainty analysis also shows the significant effect that component tolerances have on roll-off performance, often an important factor in the selection of Chebychev over Butterworth filters. The graphs also indicate the benefits to be derived from moving to higher quality components. Indeed, the analysis gives the engineer clear information on the cost/performance tradeoffs that are an inevitable factor in any filter design.

5.6 Summary

A novel method for analysing the effect of uncertainty on filter performance has been presented. Worst case performance has been mapped directly to a robust stabil-

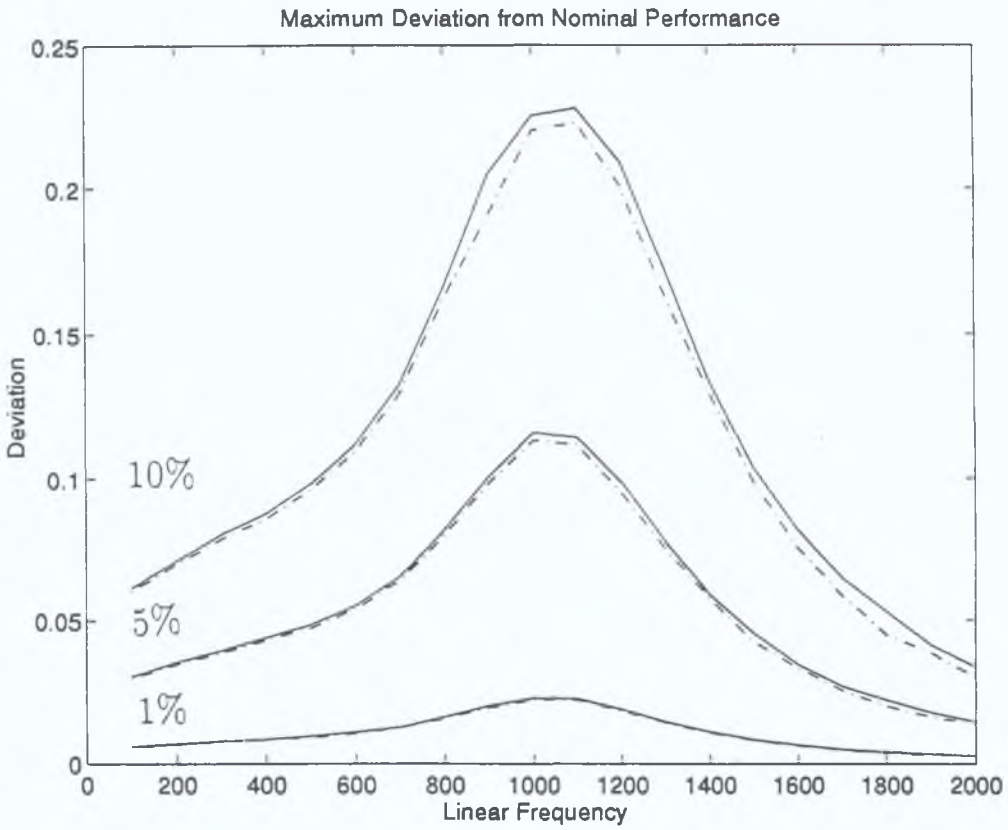


Figure 5.18: Upper(*Full*) and Lower (*Dash*) bounds on $G_{dev}(s)$ for a Chebychev filter of Order 3.

ity question, by extracting the uncertainty from the block diagram representation of the filter transfer function. This process of uncertainty extraction results in a perturbation Δ acting on an augmented *Diagonal Perturbation Formulation* of the nominal system. μ is an operator on this formulation which, given the appropriate constructions, translates directly into non-conservative worst case bounds on filter performance.

The method has been illustrated using standard Butterworth and Chebychev filters. The proposed approach provides clear frequency response information that fully addresses the interactive effects of component uncertainties. Since any linear transfer function can be expressed in terms of its DPF, the method described is applicable quite generally to all linear circuits and linear digital filters.

There are numerous other applications for such a procedure. A selection of these are mentioned at the end of this thesis.

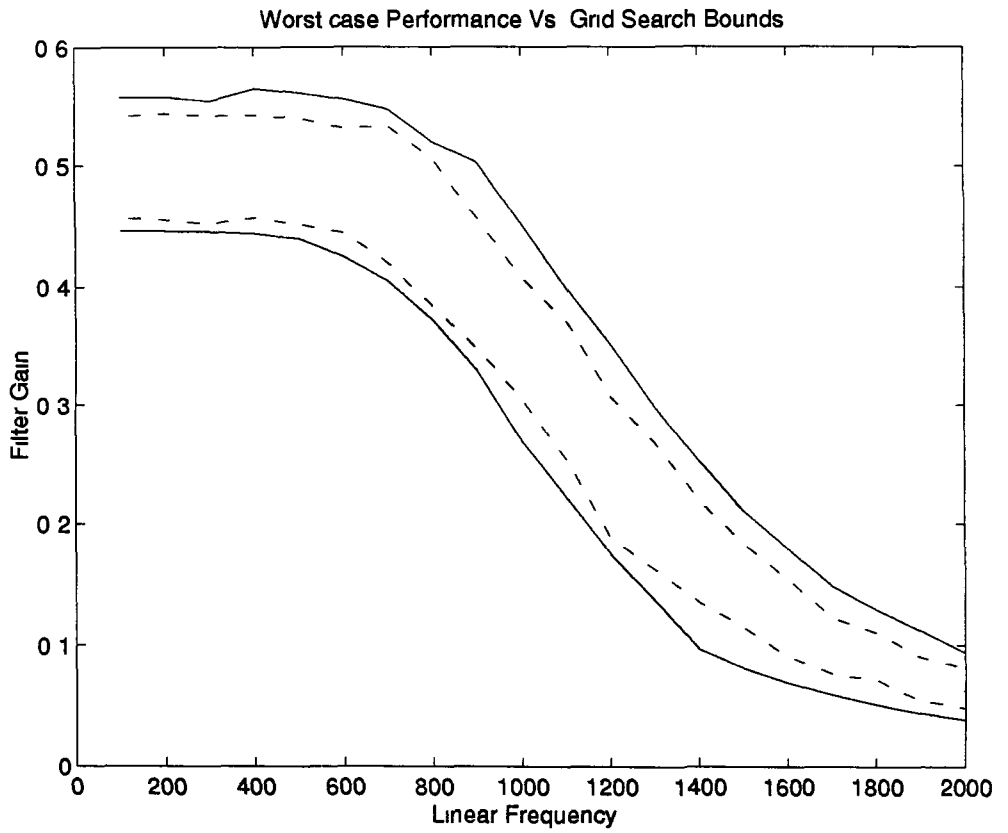


Figure 5 19 Comparison of random search (dashed lines) and μ bounds for Butterworth Filter of order 3 with 10% tolerances

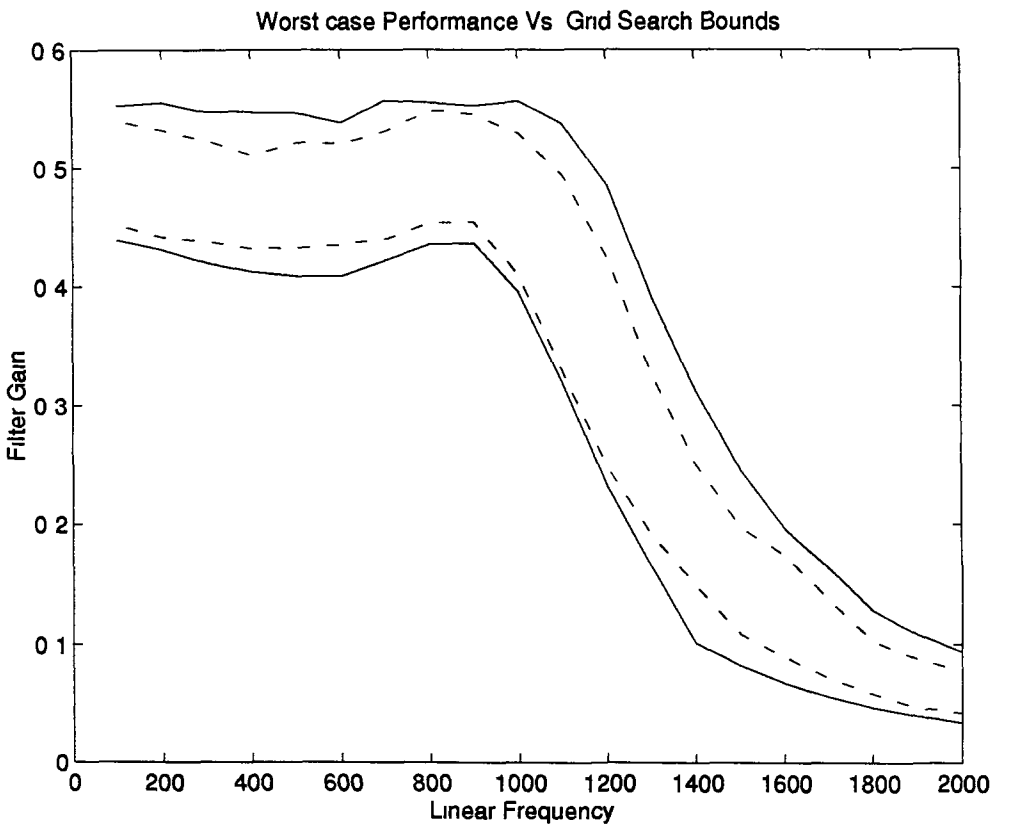


Figure 5 20 Comparison of random search (dashed lines) and μ bounds for Chebychev Filter of order 3 with 10% tolerances

Chapter 6

A New Approach to PID Tuning

PID control is the most widely used control law in industry today. It is widely accepted that PID controllers do not perform well and/or are difficult to tune in certain situations. The performance of systems that are tuned conventionally may degrade significantly when the plant is poorly modelled, is non-linear, has variable time delay and/or has many operating points. Many different approaches exist [Morari 89], [Astrom 95] that can improve the robustness of PID controllers. In this chapter a new method, based on L_1 methods, is proposed. This will allow the development of controllers that will minimise tracking errors for a family of plants. This different objective function is the main distinguishing feature of the new method. This thesis contends that such a robust time domain approach is natural for many engineering applications.

6.1 Problem Formulation

To begin, the problem is set up in a standard fashion. Fig. 6.1 shows a block diagram of the system configuration under consideration. The block P_o is the Linear Time Invariant (LTI) nominal model of the plant. This model can be a relatively crude approximation of the actual plant. The parallel combination of P_o and Δ is viewed as the actual physical plant. Here, Δ is thought of as a bounded gain dynamical subsystem that perturbs the plant away from its nominal value. A key feature of this

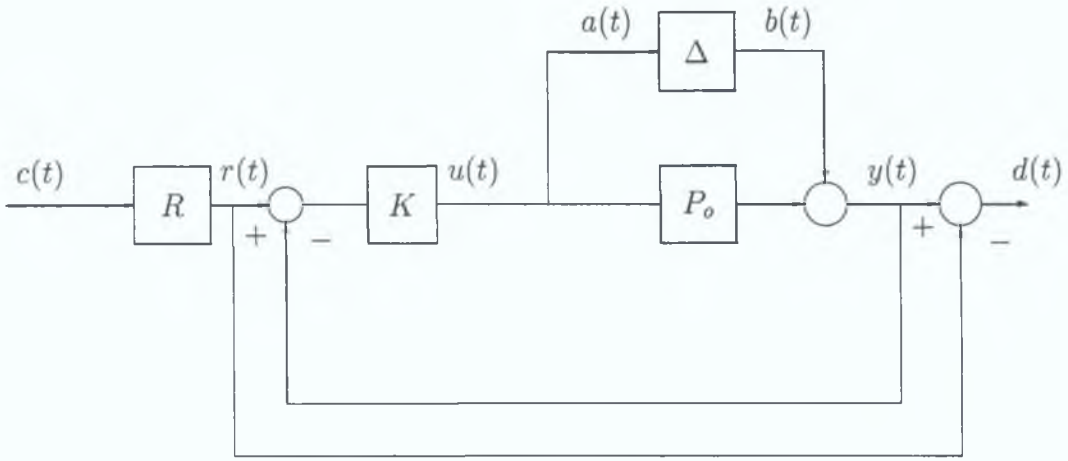


Figure 6.1: System Configuration

approach is that this perturbation may be non-linear and/or time varying. Covering the uncertainty in P_o by such a general perturbation allows difficult problems to be incorporated in the formulation.

Since this work is concerned with new tuning rules for PID controllers, the K in Fig. 6.1 will be of the form

$$K(s) = K_P\left(1 + \frac{K_I}{s} + K_D s\right)$$

The PID tuning problem is to find appropriate values for the controller constants K_P, K_I and K_D. The control objectives that are of interest throughout this chapter are that $K(s)$ should stabilise the perturbed system and yield good command tracking. To ensure that the latter performance specification can be achieved, a prefilter $R(s)$ will be required. In this way the reference input $r(t)$ of Fig. 6.1 should be considered a low pass filtered version of the command input $c(t)$. Good command tracking means that $y(t)$ should follow or track $r(t)$ closely. Thus, the error signal $d(t)$ should be uniformly small over time.

A controller K will guarantee stability *robustly* if there exists no permissible Δ which causes the corresponding perturbed system to be unstable. Moreover, it is also required that the command tracking objective must hold, not only for the nominal system, but for all systems traced out by the family of valid perturbations. This is termed a *robust* command tracking objective. Thus, a certain level of command tracking can be guaranteed for every permissible Δ.

The essential idea behind the approach is to allow the perturbation set to "cover" model limitations Best results can be expected in cases where the plant model is quite poor A PID controller is located with *optimal* immunity to the effects of Δ, taking both stability and command tracking properties into account The procedure allows rigorous determination of bounds on how well the "best" PID controller can do in terms of these objectives Hence, decisions can be made about whether PID is a suitable strategy or if a different control law is necessary to achieve the required objectives

6.2 Application of L_1 Theory to PID Tuning

Consider the formulation of Fig 6 1 Suppose that the Δ block were removed Then the 2×2 transfer function matrix from the input vector $\begin{pmatrix} b(t) \\ c(t) \end{pmatrix}$ to the output vector $\begin{pmatrix} a(t) \\ d(t) \end{pmatrix}$ can be easily computed Introduce a suitably partitioned transfer function matrix M where

$$M(s) = \begin{pmatrix} m_{11}(s) & m_{12}(s) \\ m_{21}(s) & m_{22}(s) \end{pmatrix}$$

Let $m_{ij}(t)$, $(i, j = 1, 2)$ denote the inverse Laplace transform of $m_{ij}(s)$, i e , the corresponding impulse response Let $\|m_{ij}(t)\|_1$ denote the integral of the absolute value of $m_{ij}(t)$ so that

$$\|m_{ij}(t)\|_1 = \int_0^\infty \mid m_{ij}(t) \mid \, dt$$

The element by element norms of these impulse responses can be grouped together to form the the matrix $\widehat{M}$, i e ,

$$\widehat{M} = \begin{pmatrix} \|m_{11}(t)\|_1 & \|m_{12}(t)\|_1 \\ \|m_{21}(t)\|_1 & \|m_{22}(t)\|_1 \end{pmatrix}$$

The input and output of Δ are denoted by $a(t)$ and $b(t)$ respectively From the definition of an induced norm it can be seen that for the system of Fig 6 1

$$\|\Delta\|_1 = \sup_{a(t) \neq 0} \frac{\|b(t)\|_\infty}{\|a(t)\|_\infty}$$

Therefore this quantity represents the maximum amplitude gain in the time domain of the system block Δ, maximised over all bounded inputs with the exception of the signal which is identically zero The basis for this work is in the following theorem This follows directly from [Dahleh 95] and the robust performance theorem [Stein 82]

Theorem 6.1 *With reference to Fig 6 1, the closed loop system is stable and* $d(t)$ *is bounded above by*

$$\|d(t)\|_\infty \leq \frac{1}{\alpha_2}\|c(t)\|_\infty$$

for every Δ *which obeys*

$$\|\Delta\|_1 \leq \frac{1}{\alpha_1}$$

if and only if

$$\mu_{L_1}(M) = \rho\begin{pmatrix} \alpha_1\|m_{11}(t)\|_1 & \alpha_1\|m_{12}(t)\|_1 \\ \alpha_2\|m_{21}(t)\|_1 & \alpha_2\|m_{22}(t)\|_1 \end{pmatrix} < 1$$

Here, $\rho(M)$ refers to the spectral radius of the matrix M This theorem is used to determine how well a combination of controller $K(s)$, LTI plant model $P_o(s)$ and a perturbation class $\mathcal{D}_{ex}$ will robustly track commands

L_1 theory is especially suited to this problem Like H_∞ theory an L_1 approach provides a measure of worst case system performance as Δ varies through a broad range of possibilities $\mathcal{D}_{ex}$ However, being a time domain approach, specifications like command tracking can be treated rigorously μ_{L_1}-theory provides necessary and sufficient conditions for robust stability and performance Moreover, computation of μ_{L_1} is much less expensive than standard μ for purely LTI perturbations

6.3 Description of the Approach

The proposed procedure for tuning PID controllers using L_1 methods can be described as follows -

1 Determine the maximum allowable gain of Δ, viz , $\frac{1}{\alpha_1}$ This is estimated from simulations of a representative sample of (possibly non-linear, time varying)

perturbation Δ's This quantity gives an indication of the maximum "size" of uncertainty set to be covered Choosing such a gain bound on Δ amounts to describing the level of validity of the nominal model

2 Theorem 6 1 is used to determine the worst case tracking error with the current PID controller constants Viewing α_1 as fixed and given, determine the maximum value of α_2 for which $\mu_{L_1}(M) < 1$

3 $\mu_{L_1}(M)$ is now optimised over the available design variables K_P, K_I and K_D This corresponds to minimising the tracking error $\|d(t)\|_\infty$

4 The optimised controller is again analysed for robustness using Theorem 6 1

Features

The main features of the approach can be listed as -

1 The approach treats robust performance (i e , robust command tracking) as well as robust stability, thus providing immunity to imperfections in the LTI model of the system

2 The approach handles non-linear and/or time varying perturbations Indeed, if LTI perturbations are all that can occur then standard μ theory should be used, as μ_{L_1} will be unnecessarily conservative

3 Appropriate optimisation software can be used to determine the controller which yields optimal robust stability and robust command tracking Optimally robust values for a PID controller are generally obtainable numerically If the best PID controller does not offer acceptable performance then another control law should be used

4 The necessary computations are relatively straightforward and (compared with standard μ) inexpensive

5 The approach is easily extended to include other time domain performance objectives, such as disturbance rejection

Limitations

The main limitations of the approach are as follows :-

1. The underlying optimisation problem is non-convex. Thus, standard optimisation routines are not guaranteed to yield the global optimum and may yield less favourable local minima. It is envisaged that an optimisation strategy using a genetic algorithm may be of benefit here.

2. It should be noted that $\mathcal{D}_{ex}$ is a very general perturbation set. Thus, the process of "covering" uncertainty by a $\Delta \in \mathcal{D}_{ex}$ where the only constraint on Δ is that $\|\Delta\|_1 \leq \frac{1}{\alpha_1}$ is potentially overly conservative and very demanding for a PID controller.

3. The approach is best suited to problems where the available plant models are necessarily quite poor. In a case where a precise model is available, other methods can be expected to yield better results.

4. A quantitative comparison with other tuning methods that promise improved robustness such as those in [Morari 89] or [Astrom 95] is difficult given the different objective function that is being employed in this approach.

6.4 A Second Order Example

This section illustrates the development of an "L_1-tuned" PID controller using the proposed methodology. The procedure adopted is described, and the performance of such a controller is compared with its Ziegler-Nichols tuned counterpart. As [Astrom 95] illustrates, there are a plethora of different tuning methodologies that could be chosen as a benchmark for comparison. Anyone of these can be used as a starting point from where improvement in terms of a tracking error objective function will be possible. The tuning rules of Ziegler-Nichols aim for a reduction in tracking error to 25% of the worst case value in the first period of oscillation [Ziegler 42]. This is a time domain specification and is a reasonable starting point for improvements in

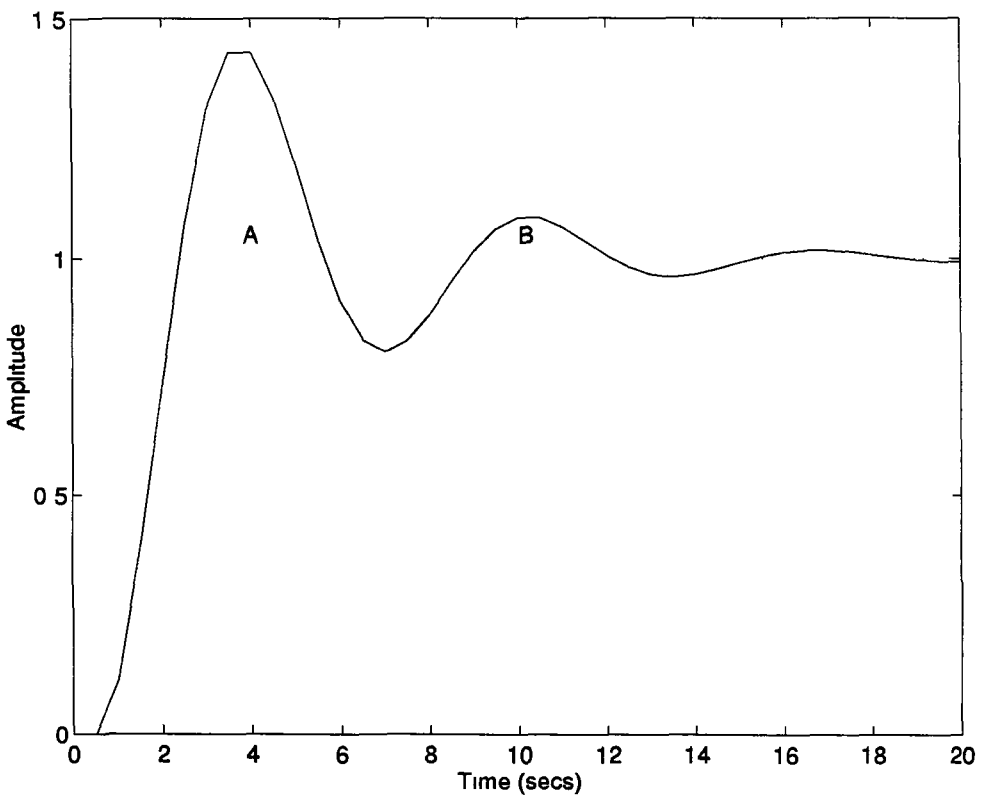


Figure 6 2 Ziegler-Nichols Tuning Objective $\frac{B}{A} \leq \frac{1}{4}$

the context of the objective at hand. Fig. 6.2 illustrates this typical rule of thumb

$$\frac{B}{A} \leq \frac{1}{4}$$

Note that this objective is not the same as that of minimising the worst case tracking error. Making improvements in this objective will necessarily compromise any properties that an earlier set of tuning rules might have.

6.4.1 Problem Outline

The system to be controlled is modelled as a critically damped second order plant with natural frequency $\omega_n = 1$ rad/sec and a damping factor $\zeta = 0.7$. The system is assumed to have a transport delay of $D = 1$ second at the plant input. Thus the nominal plant is

$$P_o(s) = \frac{e^{-s}}{s^2 + 1.4s + 1}$$

In the actual plant the value of the parameters ω_n, ζ and D are not known exactly and may deviate significantly from their nominal values. Thus the actual plant transfer function is

$$P(s) = \frac{e^{-sD}\omega_n^2}{s^2 + 2\zeta\omega_n + \omega_n^2}$$

To enable the existence of a worst case tracking error that is less than unity a pre-filter $R(s)$ is required. This was taken to be a first order low pass filter with a time constant of 2 seconds. This was selected arbitrarily and seems reasonable in terms of the nominal plant parameters.

The PID controller constants that will minimise the stated objective are now required. Thus, worst case tracking error will be minimised for a set of general perturbations. In Theorem 6.1, the only restriction on an element of this set is that its 1-norm must be no greater than the size of the worst perturbation from the grid of parameters. For this particular example, the uncertainty that is considered is LTI. Thus, the size of the maximum allowable perturbation gain may be estimated by allowing the three parameters ω_n, ζ and D to vary by $\pm$ 20% over a grid. It should be stressed that the theory allows *all* perturbations to be "covered" even if they are non-linear or time varying. When the only perturbations that can occur in practice are limited to the grid used in this example then the results that are obtained can be expected to be conservative.

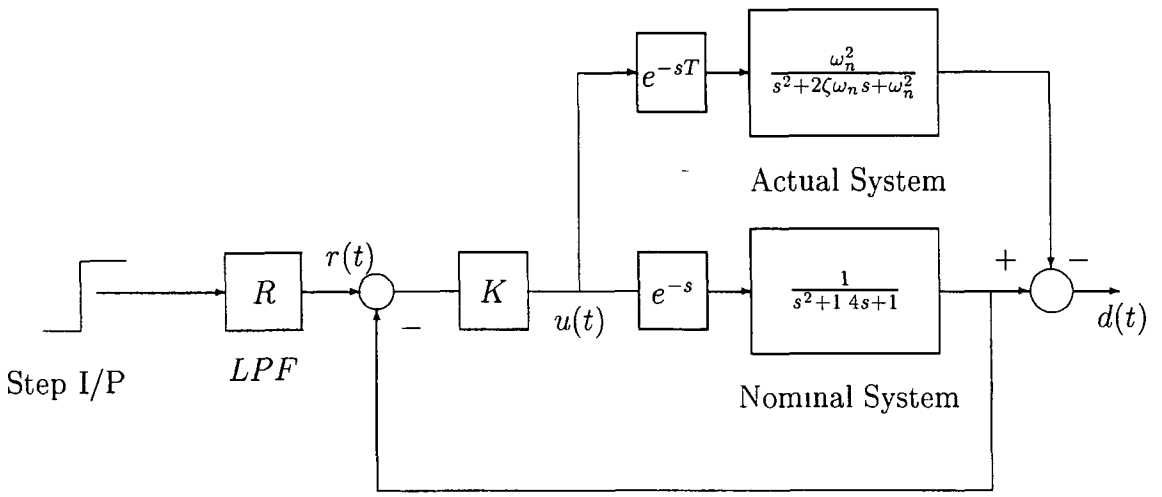


Figure 6 3 Block Diagram arrangement to determine the effect of a grid of perturbations on the nominal system

6.4.2 Procedure

The following procedure is adopted -

1 Tune the nominal system using a benchmark method Although Ziegler-Nichols tuning rules are used in this example, any other method can also be used as a starting point The ultimate gain K_U was found to be 1 98 and the ultimate period T_U was found to be 5 seconds This yields PID controller constants of

$$K_P = 1\ 98 * 6 = 1\ 18, \qquad K_I = \frac{1}{5 * 5} = 0\ 4, \qquad K_D = 5 * 125 = 0\ 625$$

2 Determine the effect of a representative sample of perturbations to the nominal system This was achieved using the block diagram arrangement of Fig 6 3 which was implemented using *SIMULINK* [Checkoway 92]

A grid of perturbations were generated by allowing ω_n, ζ and D to vary in the range $\pm$ 20% For each element of this grid the maximum amplitude of the tracking error $\|d(t)\|_\infty$ and the maximum amplitude of the plant input $\|u(t)\|_\infty$ were recorded The worst case ratio of tracking error versus plant input was recorded This gives a rough measure of the "size" of perturbations that need to be covered For this grid the worst case ratio was found to be -

$$\alpha_1 = \frac{\|d(t)\|_\infty}{\|u(t)\|_\infty} = \frac{0\ 2659}{1\ 0149}$$

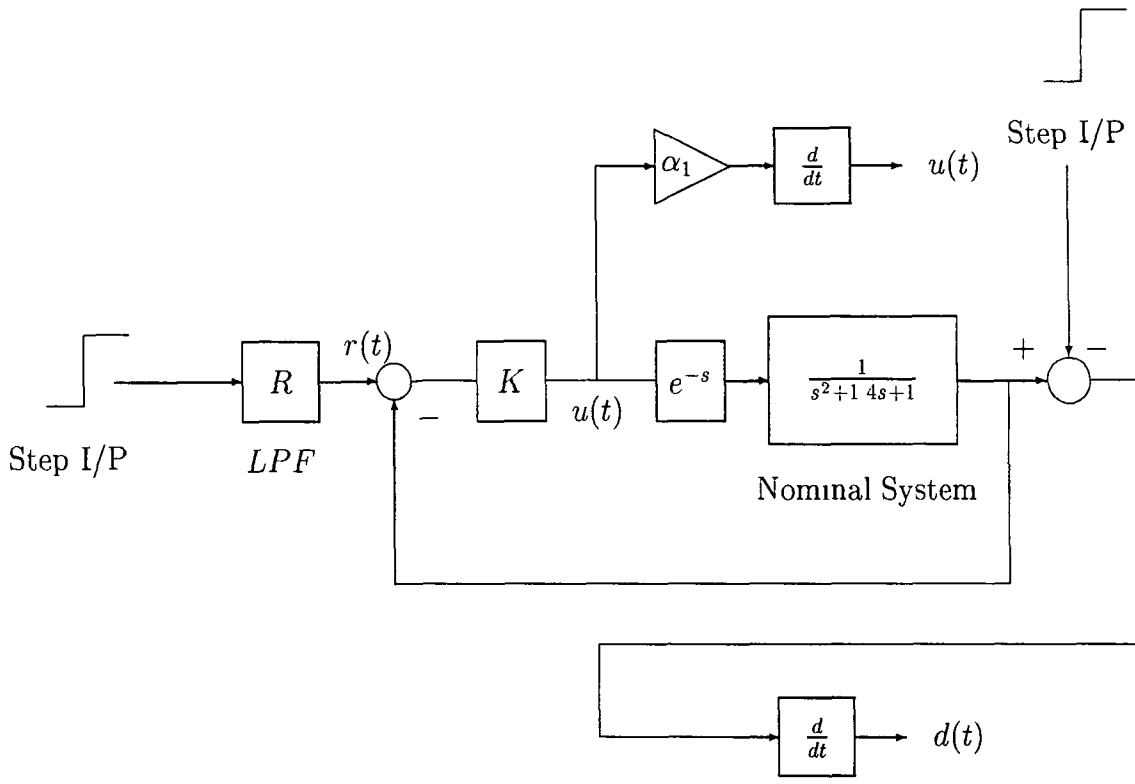


Figure 6 4 Block Diagram arrangement to determine μ_{L_1}

3 Determine μ_{L_1} for the current controller settings using the block diagram arrangement of Fig 6 4

SIMULINK determines the 2×2 matrix which consists of the 1-norms of the element by element impulse response of this system to the output signals from two perturbation Δ's, Δ_1 and Δ_2 The top row which is hit by Δ_1 corresponds to the size of system uncertainty that is being covered It is weighted by the α_1 that was computed in the previous step When this impulse response is being calculated the input from Δ_2 to the bottom row is set to zero For reasons of numerical accuracy it was more appropriate to input a suitably scaled step function to the system and then differentiate the output Δ_2 corresponds to the performance parameter α_2 is then selected, where $\frac{1}{\alpha_2}$ corresponds to a certain worst case tracking error If *SIMULINK* returns a 2×2 matrix whose spectral radius is less than 1 then it can be said that this performance objective is achievable by the current controller for the *entire* uncertainty set

4 By direct application of Theorem 6 1 a search for the PID Tuning constants

that will make the spectral radius of the impulse response matrix less than unity is performed If the spectral radius is greater than unity, the fact that the problem is linear in α_2 can be exploited so that *SIMULINK* need not be run again The spectral radius is rescaled to be less than 1 by use of a suitable α_2 This corresponds to the best possible worst case tracking error that is possible with this controller If this is not good enough the search for new PID constants continues If there exists no new controller settings that will further reduce μ_{L_1} then it can be concluded that the performance objective cannot be achieved with this control law

This process yielded L_1 tuned settings for the problem at hand which were

$$K_P = 0\,69, \qquad K_I = 0\,43, \qquad K_D = 0\,68$$

6.4.3 Results Analysis

Initially the robust stability indices can be compared, i e , $\widehat{M}_{11}$, the (1,1) element of the $\widehat{M}$ matrix is considered for 20% parameter uncertainty

Robust Stability	
Ziegler-Nichols	L_1
1 9971	1 0889

This immediately indicates that robust stability cannot be guaranteed if a PID control law is to be used and 20% parameter uncertainty is allowed Clearly, this rules out any claims being made about robust performance However, it is also clear that an L_1 tuned approach offers nearly twice the stability margin of its Ziegler-Nichols tuned conterpart Table 6 1 outlines, for the component uncertainty levels where robust performance guarantees can be made, a comparison of worst case tracking error between the two controllers The table shows that an L_1 tuned approach can buy an extra 8% of parameter uncertainty whilst maintaining robust stability

Table 6.1 Comparison of Tuning Rules for PID Controllers

	L_1		Ziegler-Nichols	
% Uncertainty	*Tracking Error*	$\widehat{M}_{11}$	*Tracking Error*	$\widehat{M}_{11}$
1	2 7259417e-01	054	4 1095409e-01	099
2	2 8429308e-01	108	4 6270461e-01	199
3	2 9744384e-01	163	5 2919897e-01	299
4	3 1248093e-01	217	6 1778391e-01	399
5	3 2973756e-01	272	7 4165950e-01	499
6	3 4977655e-01	326	9 2715831e-01	599
7	3 7335926e-01	381	1 2363977e+00	698
8	4 0145055e-01	435	1 8531002e+00	798
9	4 3562110e-01	490	3 6809706e+00	898
10	4 7796492e-01	544	3 3884844e+02	998
11	5 3180992e-01	598	-	-
12	6 0244154e-01	653	-	-
13	6 9933413e-01	707	-	-
14	8 4106776e-01	762	-	-
15	1 0667778e+00	816	-	-
16	1 4823393e+00	871	-	-
17	2 5085742e+00	925	-	-
18	9 1739832e+00	980	-	-

The performance measure shows a similar trend Fig 6 5 illustrates how an L_1 tuned controller provides much better robust performance in the first 10% of maximum allowable parameter uncertainty In the region where the parameter uncertainty is in the range [10%, ,18%] a Ziegler-Nichols tuned controller cannot guarantee robust stability

Although no guarantee that a system will be robustly stable when it is subject to an arbitrary 20% perturbation can be given, it is important to realise that this is a very exacting requirement Perturbations of this nature may not occur in practice It is now demonstrated how an L_1 approach can offer improved performance if only the perturbations from the original grid are considered The step response for the nominal

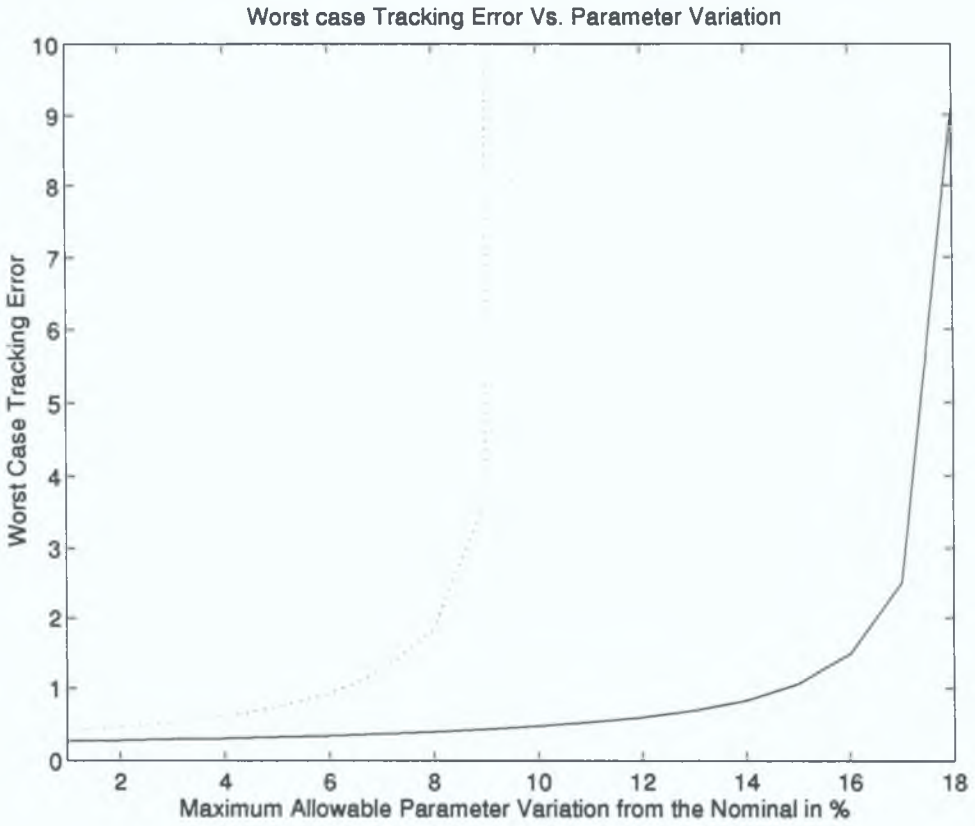


Figure 6.5: Comparison of Worst case Tracking Error with variable component uncertainty for L_1 (Full) and Ziegler-Nichols (Dashed) tuned controllers.

system together with the "worst" system from the allowed grid of perturbations is shown for the Ziegler-Nichols tuned controller in Fig. 6.6. The corresponding L_1 tuned response is given by Fig. 6.7. The L_1 tuned system provides an improved response to the worst case perturbed system from the grid without adversely affecting nominal system performance.

It is of interest to note that the process does NOT make a significant difference to the worst case tracking error that is experienced. However, in the L_1 case, the response is clearly less oscilliatory, i.e., the benefit of having a controller with a lower value for $\widehat{M}_{11}$ can be observed. This suggests that the prefilter is a significant limiting factor on further improvement in the performance objective.

To illustrate the improved stability robustness with an L_1 tuned approach a further reduction is made to the system damping factor. For $\zeta = 0.38$, Fig. 6.8 shows that the Ziegler-Nichols controller is on the verge of instability. However, the L_1 system

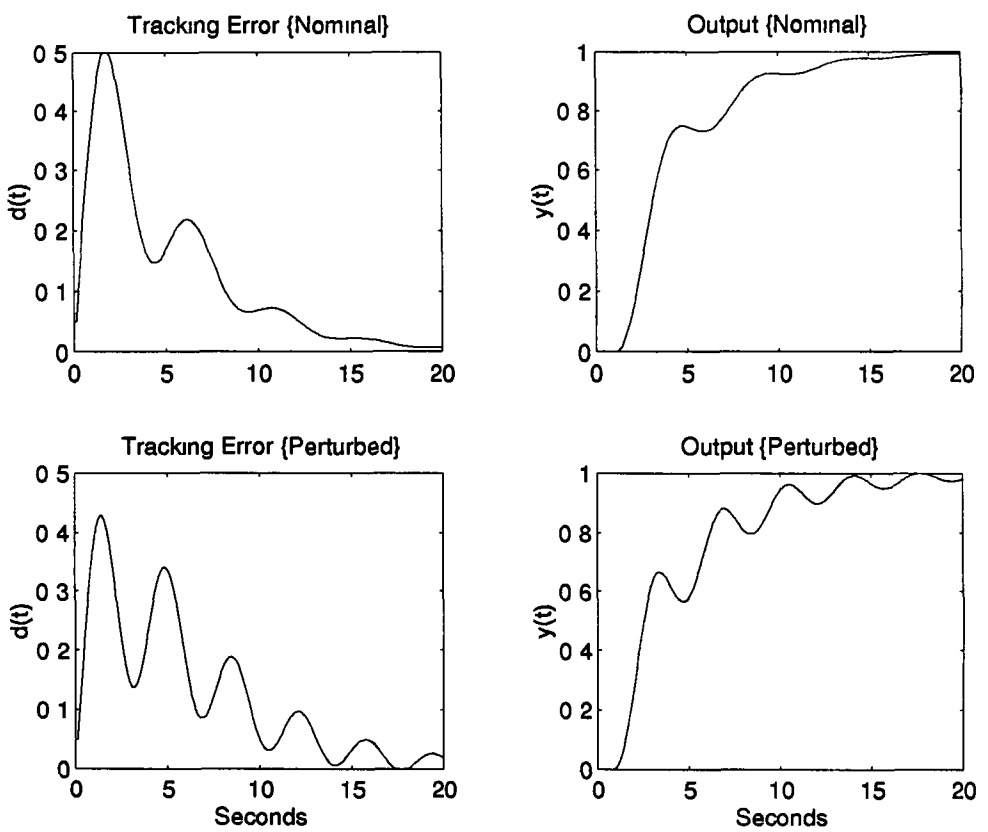


Figure 6 6 Step Response of Nominal and the Worst case Perturbed System which is tuned using Ziegler-Nichols Rules

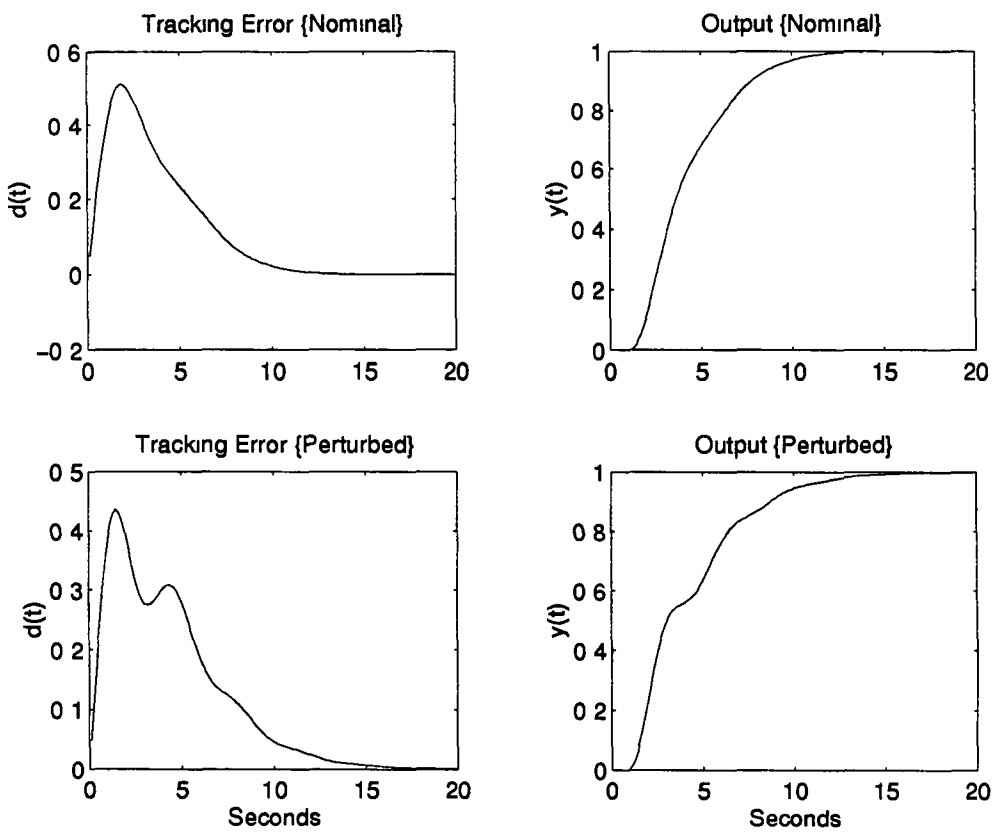


Figure 6 7 Step Response of Nominal and the Worst case Perturbed System which is tuned using L_1 Rules

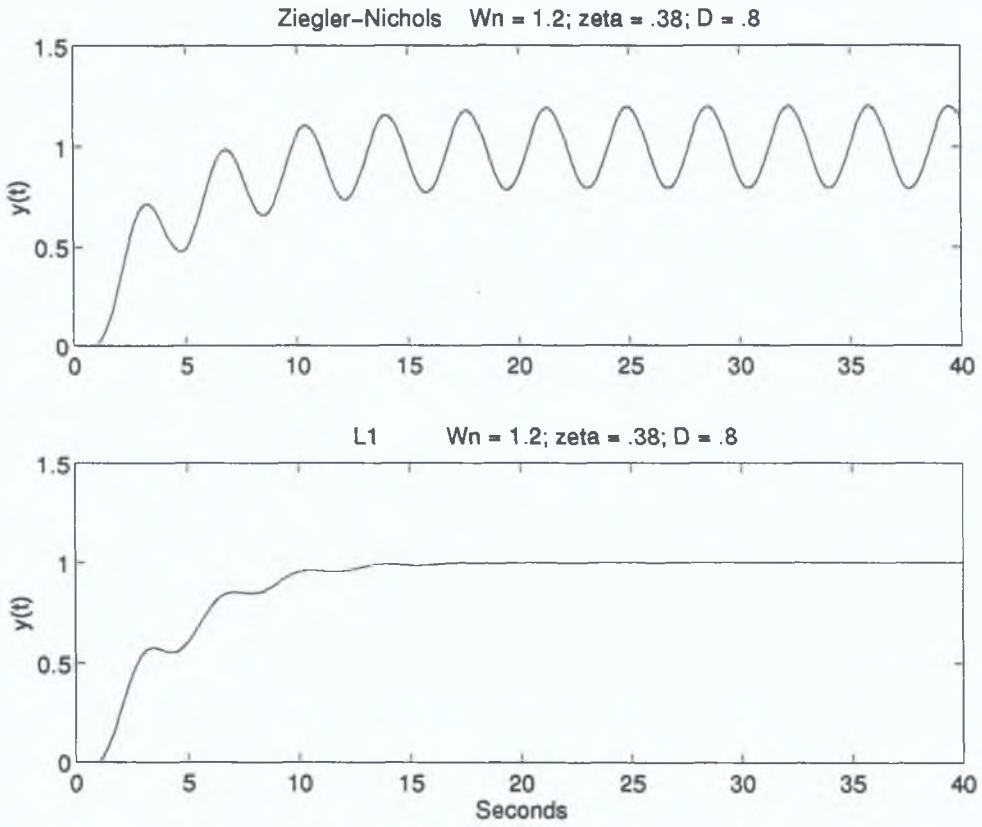


Figure 6.8: Illustration of increased Stability Robustness with an L_1 Tuned Controller.

continues to perform well. Thus it is concluded that, despite the fact no guarantees about system performance with 20% parameter uncertainty can be given, an L_1 tuned controller has better immunity to shortcomings in the plant model.

The graphs illustrate the tradeoff between robust stability and robust performance. An increase in the level of robust stability (α_1) can only be achieved by a sacrifice in the level of robust performance, i.e., a reduction in α_2. An iterative procedure based on determining whether $\rho(\widehat{M}) < 1$ for a particular combination of α_1 and α_2 quantifies this tradeoff and allows the engineer to make an informed decision.

6.5 pH Process Example

A second example is now studied which is based on the application of L_1 methods to a practical pH process. This process is a laboratory scale rig and is described at

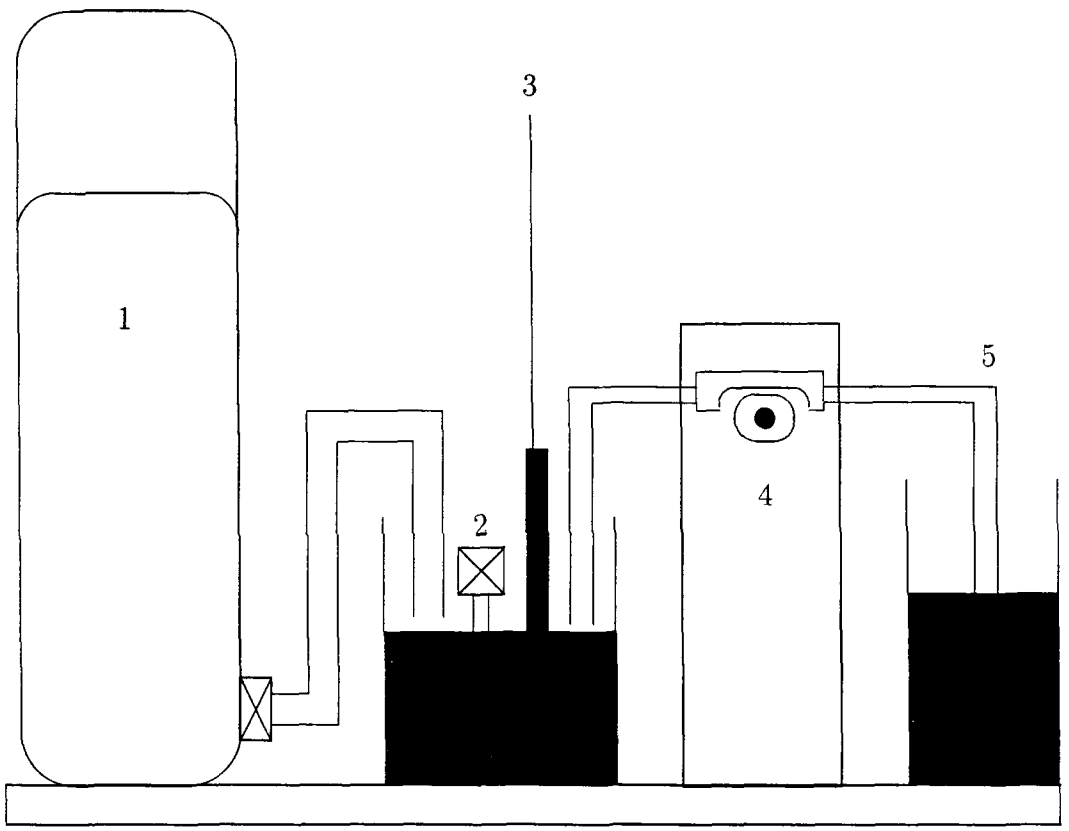


Figure 6 9 Laboratory setup for pH control 1 Liquid Tank , 2 Mixing Pump , 3 pH Meter , 4 Peristaltic Pump , 5 Acid Tank

length in [Tadeo 97]

6.5.1 Problem Outline

An illustration of the plant in question is provided in Fig 6 9 Pure water is input from a storage tank which is located on the left of the figure The flow rate of the water is a variable which is dependent on the height of water in the tank The control objective is to keep the acid concentration of the solution in the mixing tank constant at a pH of 4 0 This is achieved by the addition of tightly controlled amounts of HCl from the acid tank on the right of the figure Acid flow rate is set using a peristaltic pump The following nomenclature is used throughout this section

(a) N_d denotes the acid concentration in the tank (Output Variable, y)

(b) N_a denotes the input acid concentration (Constant)

(c) q_o is the liquid mass flow (Variable)

(d) q_a is the acid mass flow (Input Control Variable , u)

(e) V is the volume of liquid in the mixing tank

From first principles it can be shown that [Perez 95]

$$F_1(N_d, q_o, q_a) = \frac{dN_d}{dt} = -\frac{q_o N_d}{V} - \frac{q_a N_d}{V} + \frac{q_a N_a}{V} \qquad (6\ 1)$$

As can be seen this is a non-linear equation in the respective system variables Note that there is also a significant time lag in the plant between the actual tank concentration N_d and the measured value N_d^m that is fed back to the controller input This delay is approximated by a first order lag element i e ,

$$N_d^m = \frac{1}{1 + s\tau} N_d$$

Model Simplification

The following model simplifications are now considered Note that this fits in nicely with the stated aim of describing system performance using a "poor" model

1 **Constrain System Resources.** It is assumed that pure water is supplied to the mixing tank, that the input HCl is of constant concentration and that there is perfect mixing of the solution

2 **Assume a Constant Volume of Liquid in Tank.** The problem is considerably simplified by assuming that the acid and water are being pumped into a tank which is already full to overflowing This assumption, which is how the rig is set up in practice, means that $V = 63cm^3$ can be viewed as a constant

3 **View the Water Flow Rate, q_o, as a Constant.** The rate at which water is supplied to the mixing tank depends on the height of water in the storage tank

In this simplified analysis, eqn. (6.1) is viewed as a non-linear function in only two variables i.e.,

$$\frac{dN_d}{dt} = F_1(N_d, q_a) \tag{6.2}$$

Model Linearisation

The bilinear nature of eqn. (6.1) lends itself to a pretty simple linearised model. The model equilibrium points are also readily estimated using measured data. Let an asterisk beside a variable, (e.g. q_o^*), denote its equilibrium point. Following Tadeo [Tadeo 97], the generalised equilibrium position for this system is taken to be $[N_d^*, q_o^*, q_a^*]$ where

$$N_d^* = 1.6 \times 10^{-3} Mol.L^{-1}$$

$$q_o^* = 7.74 cm^3 s^{-1}$$

$$q_a^* = 0.75 * .0045 cm^3 s^{-1}$$

At equilibrium 75% of the maximum amount of acid is being added to the mixing tank. The reference acid concentration is taken to be

$$N_a = 7.08 \times 10^{-2} Mol.L^{-1}$$

It is now possible to evaluate the Jacobian for this problem. Using eqn. (6.1)

$$\frac{\partial F_1}{\partial N_d} = -\frac{(q_o + q_a)}{V}$$

which evaluated at the equilibrium position is

$$= -\frac{q_o^* + q_a^*}{V} = a = -0.1229$$

Also,

$$\frac{\partial F_1}{\partial q_a} = \frac{N_a - N_d}{V}$$

which again evaluated at the equilibrium position is

$$= \frac{N_a^* - N_d^*}{V} = b = 0.0011$$

Thus, the linearised model in state space form reduces to

$$\dot{N}_d = a(N_d) + b(q_a) = -0.1229(N_d) + 0.0011(q_a) \tag{6.3}$$

This equilibrium position is valid for a particular rate of water flow, q_o, into the mixing tank The actual flow rate of water is subject to significant variation This uncertainty can be reasonably represented as an additive perturbation to the nominal value of a Thus the "actual" value of a, denoted by a_{act}, can be given by

$$a_{act} = a + \frac{\Delta_{q_o}}{V} W_a$$

where W_a is a suitable weighting factor on the uncertain parameter Note that a normalisation based on the volume of the tank is necessary when calculating this weighting factor In a similar fashion the variation in the actual concentration of the acid in the mixing tank can be viewed as an additive perturbation to the equilibrium parameter b Thus,

$$b_{act} = b + \frac{\Delta_{N_d}}{V} W_b$$

In this way eqn (6 3) can be rewritten as

$$N_d = (a - \frac{\Delta_{q_o}}{V}) N_d + (b - \frac{\Delta_{N_d}}{V}) q_a \qquad (6\ 4)$$

which better represents the uncertainty acting on the model This, coupled with a first order measurement lag yields a pretty crude linear approximation of the pH process at hand

6.5.2 Procedure

Step 1 is to determine α_1, the maximum perturbation gain to be covered Fig 6 11 gives a block diagram representation of the linearised system that is used for this task The requisite grid of perturbations is generated by considering the following limited variation in system uncertainty

$$\Delta_{q_o} = \pm 0\ 6 cm^3 s^{-1}$$

$$\Delta_{N_d} = \pm 8\ 4 \times 10^{-3} Mol\ L^{-1}$$

$$\Delta_\tau = \pm 2s$$

Again the system was tuned initially using Ziegler-Nichols rules This yielded PID controller constants of

$$K_P = 2, \qquad K_I = 0\ 35 \qquad K_D = 0\ 1$$

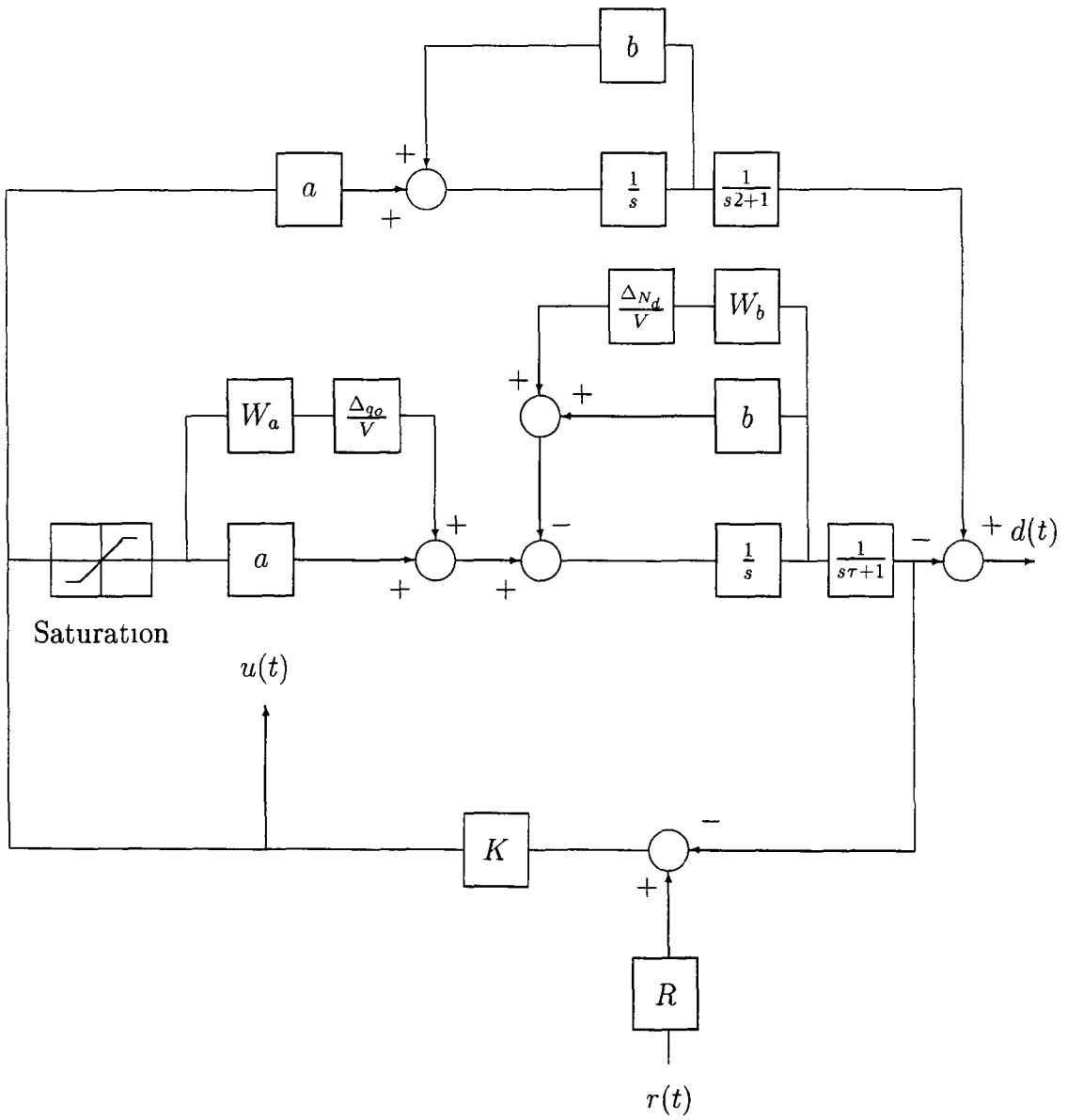


Figure 6 10 Block diagram arrangement to determine α_1 for the grid of perturbations

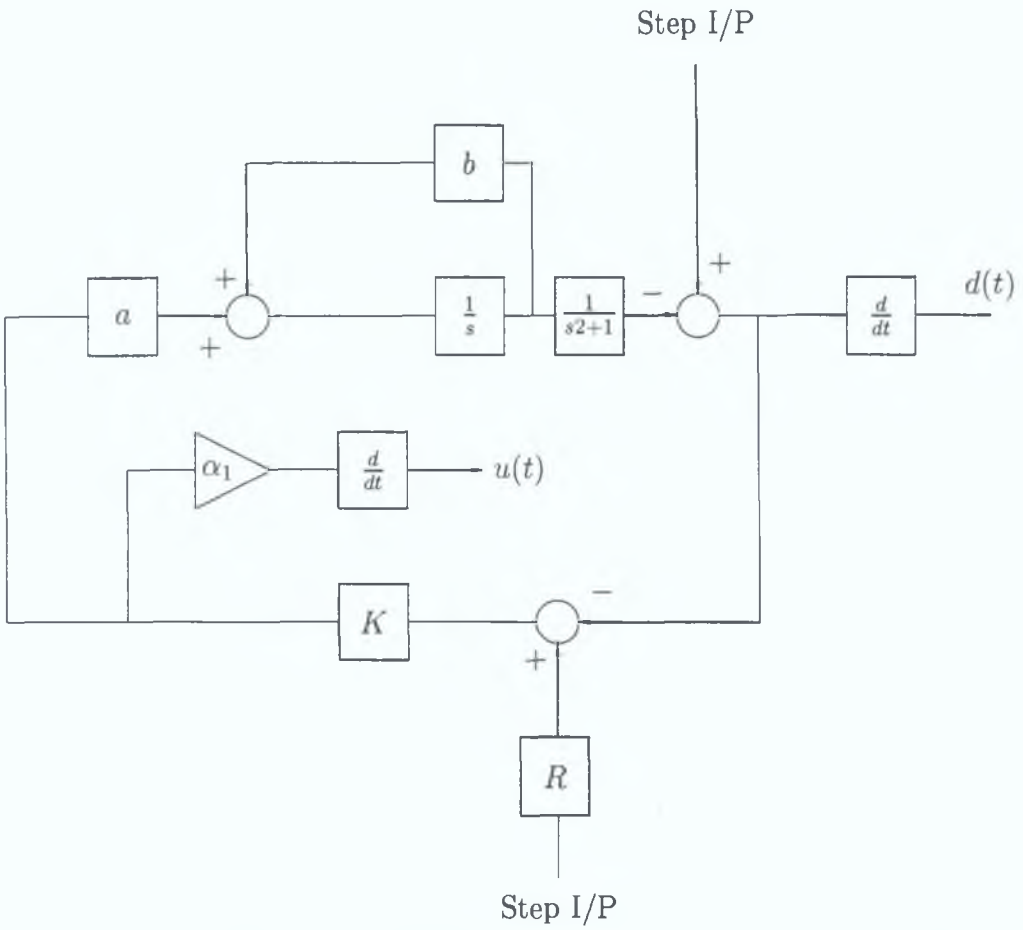


Figure 6.11: Block diagram arrangement to determine μ_{L_1} for a set of perturbations bounded in size by α_1.

The worst case output $d(t)$ and the worst case input to the perturbation $u(t)$ is again recorded using *SIMULINK*. In this case,

$$\alpha_1 = \frac{\|d(t)\|_\infty}{\|u(t)\|_\infty} = 0.3321$$

An arbitrary tracking error of 100%, i.e., $\alpha_2 = 1$, was selected for the performance step. The block diagram arrangement of Fig. 6.11 is used to determine the impulse response matrix for the linearised system. In this arrangement the saturation element has been viewed as a non-linear perturbation Δ to the linear nominal system. This is necessary for the use of a differentiator element at the output to be valid in this procedure. This adds extra conservatism to the procedure, as the family of Δ's being covered is now larger. For this example the element by element norms of the impulse

response matrix, $\widehat{M}$, and μ_{L_1} is given by

$$\widehat{M} = \begin{pmatrix} 1\,7895 & 1\,3290 \\ 2\,3276 & 1\,4423 \end{pmatrix} \qquad \rho(\widehat{M}) = 3\,3832$$

This matrix clearly illustrates that a lot of work needs to be done to guarantee robust stability let alone performance Iterating on the design variables, K_P, K_I and K_D in order to improve system robustness yields the following controller settings

$$K_P = 3\,02, \qquad K_I = 0\,09, \qquad K_D = 2\,26$$

For this vector of design variables the impulse response matrix and μ_{L_1} is given by

$$\widehat{M} = \begin{pmatrix} 0\,8024 & 0\,7920 \\ 1\,1622 & 1\,0306 \end{pmatrix} \qquad \rho(\widehat{M}) = 1\,8826$$

6.5.3 Results Analysis

As before, robustness margins are initially considered

Robust Stability	
Ziegler-Nichols	L_1
1 7895	0 8024

The stability margin associated with the L_1 tuned controller is better than double that of its Ziegler-Nichols tuned counterpart However even a modest tracking requirement like the 100% selected in this experiment cannot be guaranteed with a PID type law This reflects the demanding nature of the process and the conservative uncertainty set imposed by the design procedure The simulation work confirms the belief that only a limited amount of success can be expected with a PID control law on this problem The process suggests that it should not be hard to find a valid disturbance that will violate such a performance requirement This can be seen quite easily if a small amount of sensor noise is introduced into the model on the feedback path The design procedure suggests that the best course of action when minimising tracking error on this process is to wind down the integral element of the controller action

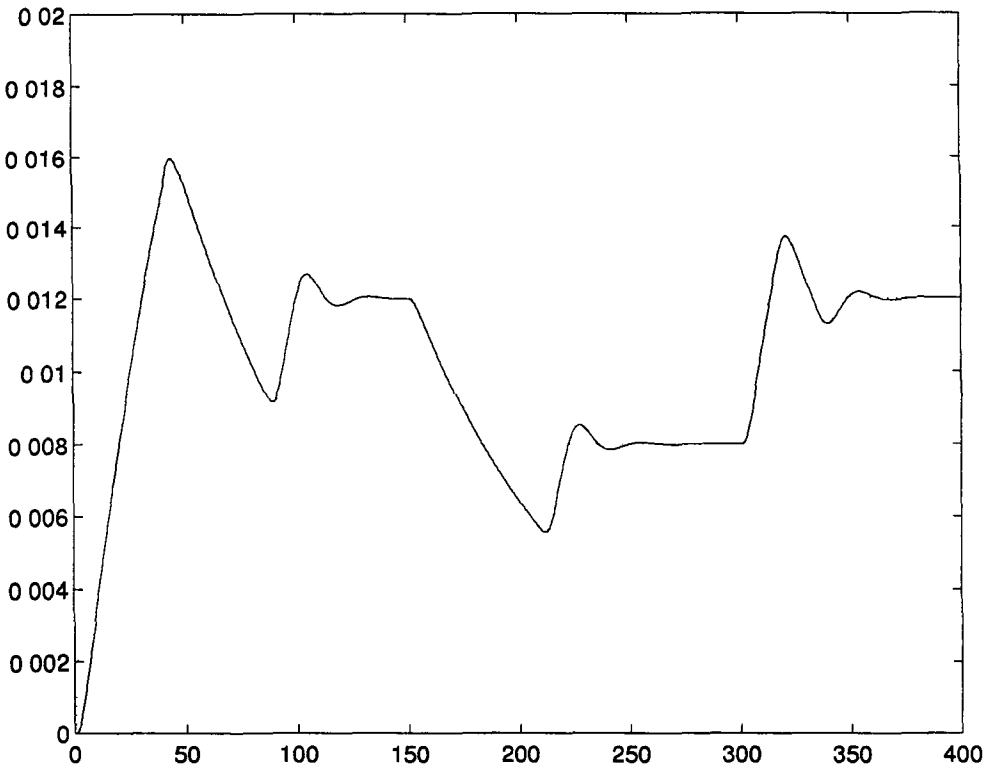


Figure 6 12 Nominal response of the Ziegler-Nichols Tuned System for the linearised pH process

The nature of the system response when integral control is used justifies this course of action in practice

Figs 6 12, 6 13, 6 14 and 6 15 show that an L_1 tuned controller does significantly better than its Ziegler-Nichols tuned counterpart when attempting to control the system subject to the grid of perturbations used in the design procedure The L_1 controller's increased stability robustness can be clearly seen with this limited uncertainty set when a programme of step changes, (the dashed lines), are input to the system For this grid of perturbations the Ziegler-Nichols tuned controller is already on the verge of instability, whilst the L_1 tuned controller continues to operate acceptably The figures clearly show how the worst case tracking error is radically improved with an L_1 tuned controller This example also shows how the design procedure provides an optimal PID controller for the performance objective at hand

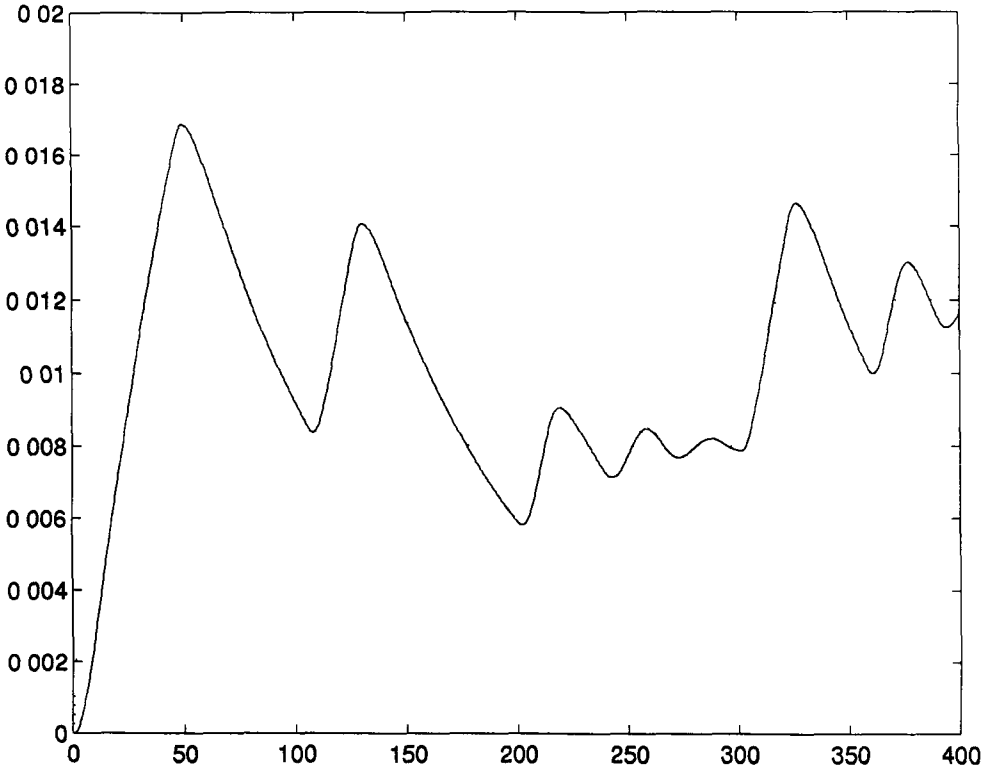


Figure 6 13 Worst case Perturbed response of the Ziegler-Nichols Tuned System for the linearised pH process

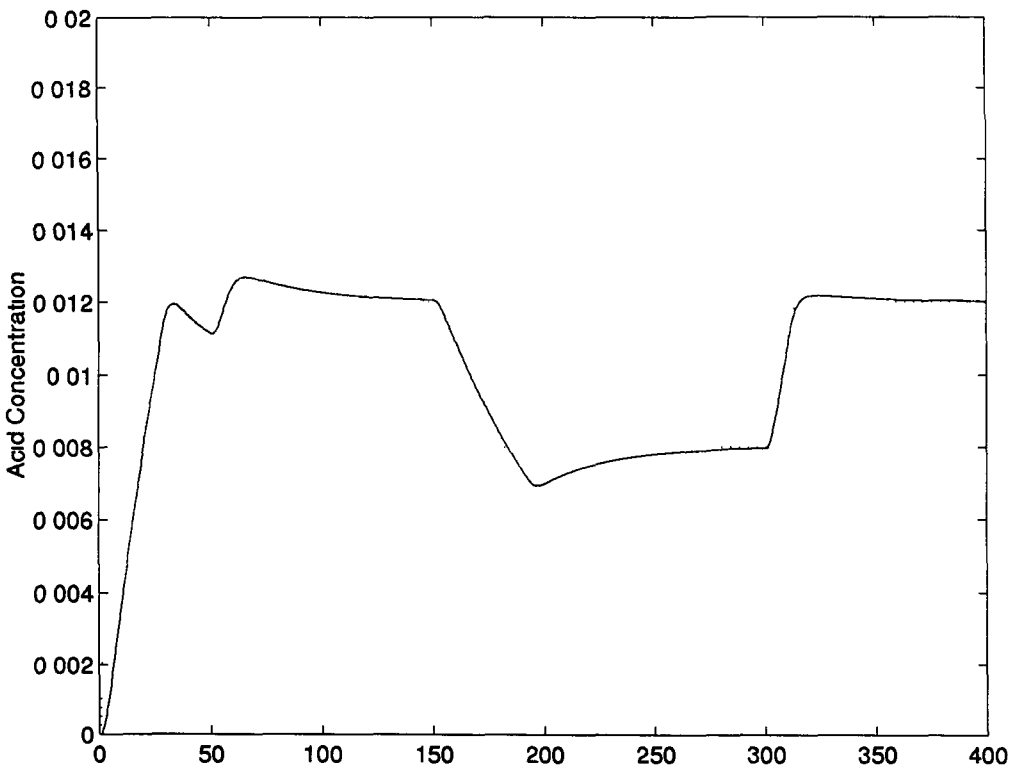


Figure 6 14 Nominal response of the L_1 Tuned System for the linearised pH process

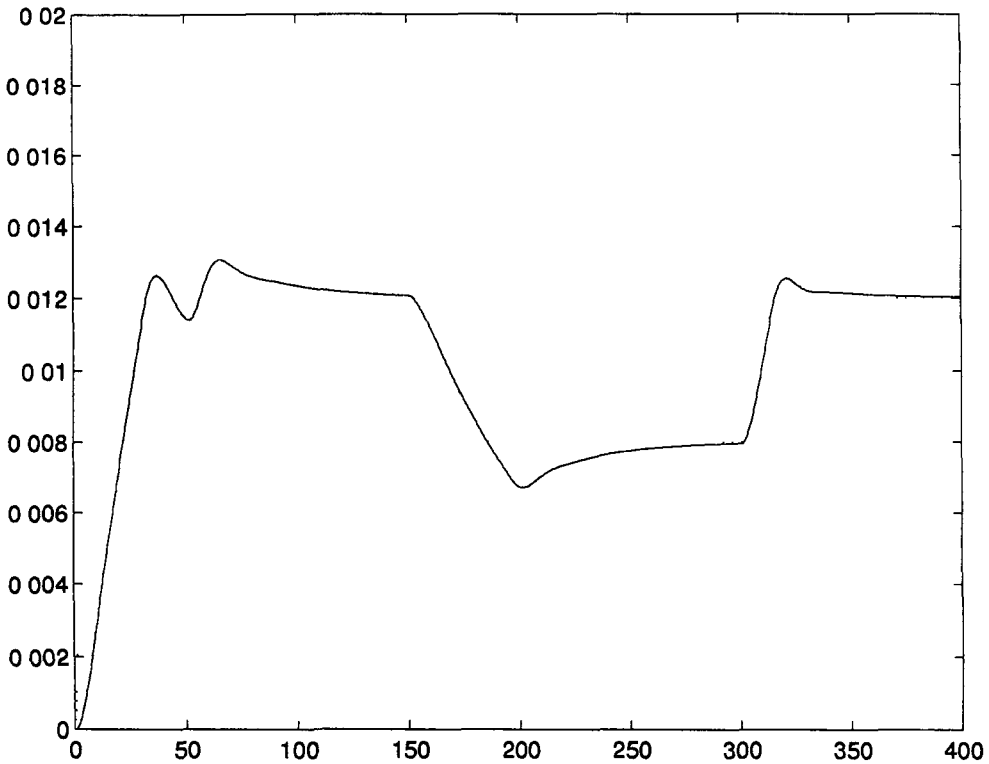


Figure 6 15 Worst case perturbed response of the L_1 Tuned System for the linearised pH process

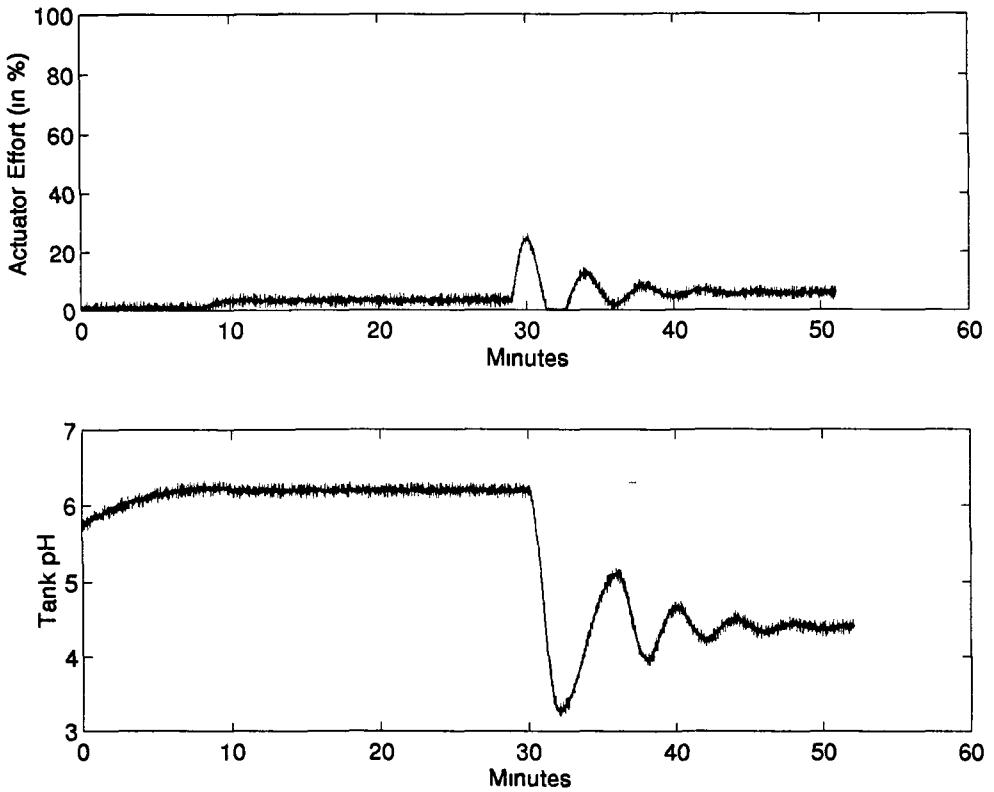


Figure 6 16 Acceptable step response for the L_1 tuned PID controller on the actual pH plant

6.5.4 Practical Validation

These PID parameters have been tested by Dr Tadeo on the actual pH plant at the University of Valladolid [Tadeo 96] It was found that, in terms of stability robustness, the L_1 design compares favourably with other PID control settings that have been used Fig 6 16 shows that a reasonable response to certain step commands close to the equilibrium point is possible However as Fig 6 17 illustrates, it is also quite easy to find a step command that will cause the system to exhibit an unacceptably large worst case tracking error These graphs show that a PID approach is not the correct solution to this problem A full μ-synthesis approach is outlined in [Tadeo 97]

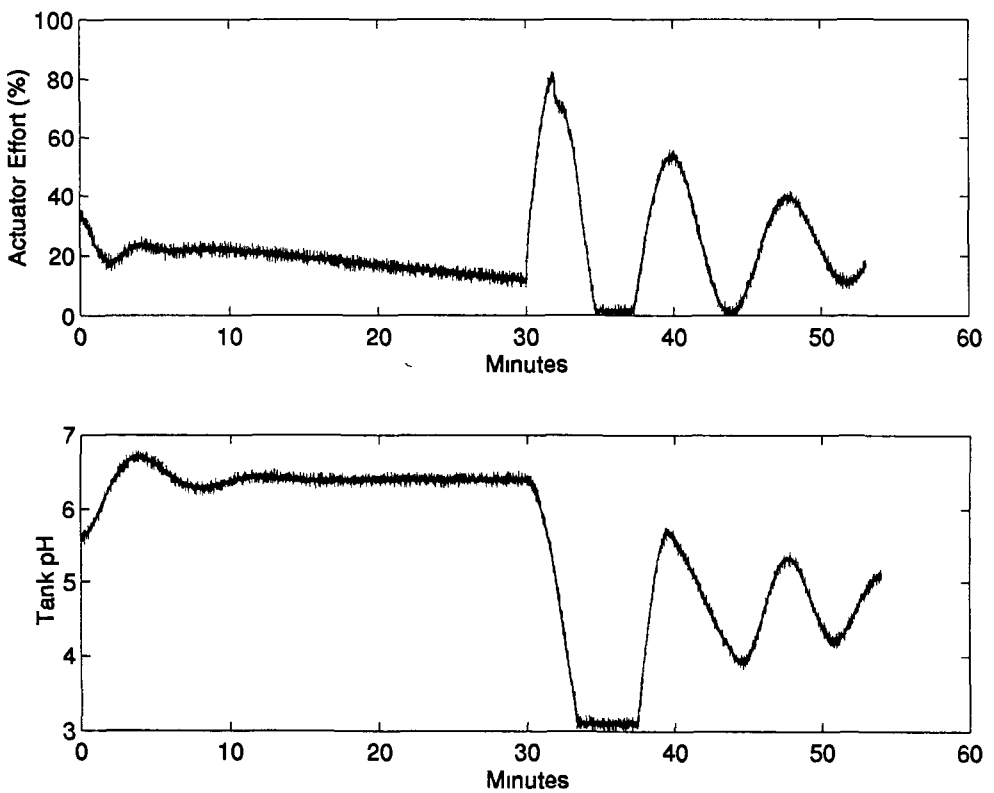


Figure 6 17 Unacceptable step response for the L_1 tuned PID controller on the actual pH plant

6.6 Summary

A new methodology for evaluating and tuning PID controllers has been proposed which is based on L_1 methods The L_1 theory allows convenient evaluation of maximum command tracking error for a given combination of controller settings, nominal plant and gain-bounded uncertainty class The new approach provides a rigorous way of assessing how well PID controllers achieve robust command tracking and robust stability

The main features of this new method are that -
(i) It is an easy to use, computer based, design approach which does not require the engineer to be a robust control specialist
(ii) The approach generally yields controller settings that gives the best possible *robust* performance when attempting to meet time domain control objectives
(iii) It takes full and due account of the (substantial) limitations of the plant model, i e , of the fact that P_o is a relatively crude approximation of the actual plant

The efficacy of the method when plant parameters are subject to wide variation and/or substantial uncertainty has been illustrated by examples at a computer simulation level When used to tune PID controllers, the new method can be used to improve the robustness of this most popular of control laws provided a correct choice of objective function to be minimised is made The procedure has also been applied to a practical non-linear pH process with some success on a narrow grid of perturbations However, the new procedure clearly demonstrates that a PID control law is not suitable for this process given the performance requirements and the system non-linearities that are involved

The breadth of the uncertainty set that μ_{L_1} covers compared with standard μ for purely LTI perturbations can be seen as both a strength and a weakness While it is true that μ_{L_1} can offer reasonable results with quite poor system models, it is also a fact that such a general uncertainty set can place unreasonable requirements on controller performance This conservatism can be clearly seen in the pH process example where it is impossible to obtain a PID controller that will promise any kind of reasonable performance It is clear that further work is necessary to reformulate the problem in such a way that the uncertainty sets which are used would be more

reflective of the actual perturbations that a system can be subjected to In this way conservatism will be minimised as a controller will not be asked to provide robustness guarantees against perturbations that cannot occur in practice

Chapter 7

Conclusions

This thesis has developed a new way of estimating μ_{co} that is reliable, efficient and global in character. The new approach compares favourably with existing commercial code in terms of the quality of bounds that are generated and the range of performance questions that can be dealt with. New applications for μ theory have been developed. μ has been shown to be an appealing way of determining the worst case effect of component uncertainty on filter performance. Finally, a μ-like theory has been developed which allows the time domain performance of PID controllers to be determined in a rigorous manner. This chapter reiterates the main findings of this work and concludes with a look at some possible future research directions.

7.1 Development of New Algorithms for the Computation of μ_{co}

Throughout this work the convex estimate of μ, denoted here by μ_{co}, has been studied since in full generality μ is not computable. It has been shown that the geometric form of the Hahn-Banach theorem can be applied to the problem of computing μ_{co}. This theorem provides the basis for an optimisation problem that is easily understood without the introduction of a large amount of new terminology. In essence, the question of robust stability reduces conveniently to whether 0 is an element of a

certain set. This counters a popular criticism of the μ approach to robust control, namely its inaccessibility.

Several new algorithms for the computation of μ_{co}, all using the Hahn-Banach theorem, have been presented. This thesis has demonstrated the clear superiority of dual methods over a corresponding primal approach when computing μ_{co}. In addition, the dual method which uses a 1-norm based optimisation strategy has distinct advantages over its 2-norm based analogue in terms of accuracy and reliability.

Convergence

An algorithm which uses the Hahn-Banach theorem to compute μ_{co} requires an inner loop and an outer loop. The inner loop determines whether a separating hyperplane between a convex set and 0 can be found for a certain level of system uncertainty α. The outer loop selects suitable candidate values of α. In its basic form a binary search can be used to select these candidate values. A binary search guarantees convergence for the outer loop. Convergence has also been demonstrated for the inner loop problem. The existence, or otherwise, of a separating hyperplane can always be ascertained. Hence the algorithms that have been developed will converge to μ_{co}.

Improvements to the Basic Algorithm

Improvements have been made to the basic algorithm which reduce computing times significantly. These changes affect both the inner and outer loops of the algorithm. The outer loop changes focus on the selection of new bounds on μ_{co} which improve, or at least do no worse than, a basic binary search approach. These new bounds can be divided into hard, rigorous, upper bounds on μ_{co} and soft bounds which are known to be good choices for the next iteration of the inner loop.

Hard bounds on μ_{co} can be obtained from an application of a prescaling procedure due to Osborne which minimises the condition number of a candidate matrix. Good bounds are also possible from an application of results in matrix pencil theory. Matrix pencil theory introduces transformations that change the size but preserve the sign of

the eigenvalues of a matrix Thus, given a system gain that ensures robust stability, i e , a confirmed upper bound on μ_{co}, the use of pencil theory allows a new, tighter upper bound on μ_{co} to be generated Pencil theory is only applicable to problems where the uncertainty set is constrained to be complex Soft bounds on μ_{co} can be computed during the inner loop These are deduced from an analysis of vectors close to the boundary of the convex set in question These bounds are not rigorous since they are based on the nature of the current best separating hyperplane The existence of a hyperplane that may invalidate claims made on the basis of these tight constraints is a possibility Hence, this analysis yields bounds that are close to, but which may be on either side of μ_{co}

Improvements are also possible in the performance of the inner loop itself A significant software overhead can be reduced during computation if the constraints used in the generation of the first linear programming tableau at the start of an iteration are known to be good This can be achieved if selected good constraints are stored after each iteration of the inner loop The concept of a good constraint is clearly a dynamic one that needs to be reviewed after each iteration of the inner loop Judicious use of constraint pruning based on only keeping good constraints significantly reduces computing times

Appropriate changes to the basic form of the algorithm have been made so as to preserve convergence At worst, the outer loop in an improved form of the algorithm can use a binary search as a starting point From here other candidate bounds are accepted or rejected on the basis of whether they are better than the binary search bounds which are already in existence The convergence properties of the inner loop are unchanged by analysis and storage of good constraints Thus, it can be concluded that the inner loop is still convergent

7.2 Validation of and Computational Experience with the New Algorithm

The new algorithm has been validated using a wide range of matrices which have been generated using several different strategies The basic validation has been achieved

with problems that are pseudo-randomly generated using a construction due to Fan This construction ensures that the candidate matrices have $\mu_{co} = 1$ The algorithm has also been tested successfully on a large number of completely random and practically motivated, purely complex and mixed uncertainty problems

In all cases, the new algorithm guarantees the location of bounds on μ_{co} within an arbitrary distance of each other Analysis of the improved form of the new algorithm has shown that computation times are competitive, albeit slower, than existing commercially available code for the computation of μ Bounds that are accurate correct to *MATLAB*'s operating precision have been demonstrated on arbitrary problems using the new algorithm In addition it is always possible to calculate these bounds in a reasonable period of time

The new algorithm succeeds in offering improved flexibility on the type of performance questions that can be posed Parametric, block and repeated uncertain elements can be dealt with that are independent or dependent on a given performance scalar In this respect the new code has a greater range of applicability than the well known μTools software The new algorithm also improves on μTools in terms of the accuracy and reliability of the bounds on μ_{co} being calculated This is particularly true when analysis of a mixed uncertainty problem is necessary The convergence properties of the new algorithm guarantee that bounds on μ_{co} can be computed to within user defined tolerances

Limitations in linear programming performance are acting as a barrier to computation times For a Simplex LP solver, difficulties begin to arise when problem sizes become larger than 7 This is a major restriction on the applicability of dual algorithms to practically motivated problems One solution may be to use an interior point LP solver to determine μ_{co} While this improves computation times, an exponential rise in FLOP requirements is still observed with increasing problem size The interior point code used in this work is recognised as the best that is currently available However, it has also demonstrated reliability problems The use of a Simplex algorithm with appropriate anti-cycling software guarantees that a solution will be found In order to preserve convergence properties of the new algorithm, it is considered appropriate to stick with a Simplex LP solver, at least for the time being

The General Proximity Problem

The use of the Hahn-Banach theorem has meant that a lot of time has been spent estimating whether 0 belongs to a convex set. The convergence properties of the algorithm often dictate that this set is configured so that 0 is *extremely* close to its boundary. The computational experience gained during the course of this work justifies some comments in this area that are of general interest. A 1-norm dual approach has been shown to be, in general, superior to its 2-norm based primal and dual counterparts when attempting to solve this proximity problem. A 1-norm dual approach allows set membership decisions to be made whose accuracy are limited only by *MATLAB*'s operating precision. This level of accuracy is orders of magnitude better than that which is possible using a 2-norm primal or dual approach.

7.3 Worst Case Filter Sensitivity With Uncertain Components

A novel method for analysing the effect of uncertainty in component values on filter performance has been presented. This has been achieved through novel applications of the robust performance theorem. Worst case filter performance can be assessed by extracting the uncertainty from the block diagram representation of the filter transfer function. This process of uncertainty extraction results in a perturbation Δ acting on an augmented *Diagonal Perturbation Formulation* of the nominal system. Analysis of the limitations that stability requirements place on such a formulation allows the effect of uncertainty on filter performance to be completely characterised. The structured singular value is an operator on the *DPF* which translates directly into non-conservative worst case bounds on filter performance.

Separate robust stability questions have been formulated which allow different effects of uncertainty to be ascertained. The basic *DPF* creates a feedback loop which allows $G_{max}(s, \Delta)$ to be determined. $G_{max}(s, \Delta)$ is the maximum possible gain of a filter when components are allowed to vary within prespecified levels. Subtraction of the nominal response from the basic *DPF* results in the determination of $G_{dev}(s, \Delta)$. $G_{dev}(s, \Delta)$ represents the maximum Euclidean distance that uncertain components

can perturb a nominal filter response to on a polar plot. Hence, $G_{dev}(s, \Delta)$ also provides information on the phase effects of component uncertainty. Inclusion of $G_{max}(s, \Delta)$ on the feedback path of a suitably defined system loop allows the computation of $G_{min}(s, \Delta)$. $G_{min}(s, \Delta)$ represents the minimum possible gain of a filter when component values are allowed to vary.

Any linear transfer function can be expressed in terms of its *DPF*. Thus, this procedure can be applied to any circuit which exhibits a linear frequency response. The method has been illustrated using Butterworth and Chebychev filters. Ladder network realisations are particularly good applications for this process because they tend to have a large number of similar components. Two port representations of each different component can be cascaded so as to construct the final filter in a structured manner. This permits the construction of software which can assess the effect of uncertain components on a standard filter of arbitrary order. The proposed approach provides clear frequency response information that fully addresses the interactive effects of component uncertainties. The results gathered clearly show the effect on a design procedure of moving from 10% to 1% ratings on permitted component variation. This provides rigorous, repeatable data which can aid the designer making an informed choice about component selection.

7.4 Development of New Tuning Rules for PID Controllers

A new way of evaluating and tuning PID controllers has been proposed in this thesis which is based on L_1 methods. The new approach relies on viewing, possibly substantial, modelling errors associated with a system as perturbations. These perturbations can be non-linear and/or time varying. The only restriction is on the size of these perturbations. By specifying the size of the allowed perturbation it is possible to "cover" all uncertainty using a single Δ.

An application of a time domain based robust performance theorem means that an evaluation of maximum command tracking error for a given combination of controller settings, nominal plant and gain-bounded uncertainty class is now possible. This

requires the introduction of the μ_{L_1} operator. Although different from standard μ in that it is time domain based, it serves the same purpose as a measure of stability margin for robust performance questions. The breadth of the uncertainty class that this operator supports can be seen as a double edged sword in that a limitation of the approach is its conservatism. As yet, μ_{L_1} does not support the use of engineering judgement in limiting the types of perturbation that a system will be exposed to in practice.

However, this new approach does possess many attractive features. The computation of μ_{L_1} is straightforward, certainly much less taxing than standard μ. The approach is easily used and does not require the engineer to have a large amount of specialist knowledge. The new approach provides controller settings which yield the best possible tracking error for the combination of linear nominal system and gain bounded perturbation class. A linearised nominal system can quite possibly be a pretty crude approximation of the actual plant. The new approach also takes full and due account of any limitations in the plant model.

Validation of the Approach on a Second Order System

A validation of the approach has been completed at a computer simulation level. A second order system which is subject to substantial parameter uncertainty and time delay variation has been defined. The "best" possible PID controller which will minimise tracking error over the nominal and *entire* set of perturbed systems has been found. The process allows a look up table of worst case tracking error versus allowed % parameter uncertainty to be generated. Provided the objective function to be minimised is chosen correctly, these new tuning rules can quantify and improve the robustness of this ubiqutous control law.

PID control of a pH process

This new tuning procedure has also been applied to a practical non-linear pH process with some success on a narrow grid of permitted perturbations. The pH process has been linearised about equilibrium points that have been determined from test data. Application of the new procedure shows that it is relatively straightforward to de-

termine a level of uncertainty that will guarantee robust stability. This is achieved by imposing large restrictions on the size of the perturbations to the nominal model. However, despite the existence of a large stability margin it is not possible to find controller constants so that $\mu_{L_1} < 1$ even when the tracking error objective is merely to within 100%. The breadth of the uncertainty set covered by μ_{L_1} renders it impossible to guarantee any kind of a reasonable tracking error using a PID control law. Whilst a PID law is not suitable for this problem, the application of this new procedure clearly demonstrates the need to reduce conservatism by a better description of the real disturbances that act on the system. A benefit of this new approach is that precise information about just how well a PID law can or cannot do is available to the engineer in the design phase of a project.

7.5 Future Research Directions

To conclude, a few comments are made about possible avenues for further research. These are grouped into four principal areas.

1. How algorithm performance can be improved.
2. Further application areas for the analysis of component uncertainty.
3. How conservatism can be reduced in the PID tuning work and how the search for the "best" controller can be improved.
4. The need to make μ more accessible.

7.5.1 Algorithm Performance

The validity and reliability of the new algorithm have been demonstrated in this work. However, more work is required to improve computation times. This is particularly true with larger problems. It is fair to say that comparison with a finely tuned piece of commercial software is a stiff task. However, more than 50% of the new algorithm's time is spent inside an LP solver. Dramatic improvements in computing

times should be possible when the linear programming difficulties that have become apparent during the course of this work are resolved

For an LP solver based on the Simplex method the best approach would seem to be to further reduce the number of constraints that need to be handled This would require an analysis of constraints during the execution of an inner loop It should be noted that the number of variables under consideration remains constant during an inner loop iteration A reduction in the number of constraints would therefore seem to be possible by consideration of the dual form of the linear programming problem in question Another possibility is to actually get inside the simplex solver itself and see whether the number of constraints being used can be reduced "on the fly"

Another avenue would be to study interior point methods to see whether the reliability problems that have been observed can be addressed It is noted that computation times, while better, are also still an issue for interior point LP solvers While μ_{co} is regarded as a good estimate of μ, the example studied during this work shows that a significant gap remains, particularly for practically motivated examples This gap reduces the room for manoeuvre that an engineer has in controller design A branch and bound strategy, which would consider μ_{co} for suitably defined subsets of the particular uncertainty class in question will reduce the conservatism associated with a convex estimate

A similarity also exists between the problem addressed by the new algorithm and the solution of certain Linear Matrix Inequality (LMI) problems It will be interesting to see how the performance of the algorithm compares with existing solutions to suitable problems

7.5.2 Analysis of Component Uncertainty

This thesis has shown how μ can help the designer make an informed decision about the effect of moving from components with 10% tolerances to their 1% counterparts The procedure presented in this thesis can be expanded so as to provide the designer with complete differential sensitivity information about particular filter implementations

There are numerous different application examples where exact information of just how badly a frequency response can deviate from nominal behaviour can be really useful. Audio applications provide perhaps the best case in point. Audio buffs pay exorbitant sums for minute perceived improvements in sound quality. Analysis of audio amplifiers using μ will yield an exact measure of the payback involved in using high quality components. These techniques can also be used to yield the error margin associated with the use of linearised models of transistor behaviour. This will be of particular benefit to designers who need to know exactly how wrong simulations of transistor behaviour using a package like *PSPICE* can be.

Further work is necessary to investigate the feasibility of automating the whole process of uncertainty extraction and *DPF* generation. It would seem reasonable that an object oriented approach might yield the best results. This would mean defining a system element in terms of a standardised set of input/output equations. The process of cascading these elements should be simplified using this process.

7.5.3 PID Tuning Work

There is a need to develop mathematical operators that can reduce the gap between the restrictive LTI uncertainty set that is handled by μ and the restriction free uncertainty set that μ_{L_1} caters for. At the moment the performance requirement that μ_{L_1} places on a controller is very severe, due to the broad scope of this uncertainty set. It is necessary that the theory develops so that the engineer is given the flexibility to tightly specify the perturbation set that is likely to be incident on a nominal model.

The search for the "best" PID controller that will minimise worst case tracking error is non-convex. It is possible that the gridding methods that have been used thus far to determine controller constants have yielded local rather than global solutions. One alternative is to use an improved search strategy using a genetic algorithm which could well overcome this limitation.

In order to move beyond the simulation level there is a need to improve on the preliminary practical data that has been collected thus far. The L_1 theory has been developed for the sampled data domain so no new point of principle would seem to arise through the use of digital controllers. The simulations have proved quite

sensitive to sensor noise so an investigation of the extra filtering requirements that are required during a practical implementation would seem to be necessary

7.5.4 Accessibility

Further work is also necessary on the user interface to the software The focus thus far has been on designing functional code It should be emphasised that this work is not purely cosmetic It is important that the code is flexible enough to deal with a variety of performance questions without the engineer having to engage in a large learning curve For example, this would be seen as a major drawback of the μTools software μ will only gain acceptance when the support tools are accessible

Judging by the literature, there is also a need for more application work using a μ approach Acceptance of μ will be accelerated by comprehensive comparitive analysis with say, fuzzy or adaptive strategies on benchmark problems that can be easily repeated One example of this would be a flexible beam problem similar to that mentioned in [Doyle 92] This problem can be constructed so as to be relatively challenging quite inexpensively Each type of design strategy will have strengths in different areas More real problem data is required to rigorously define what each of these strengths will be

Bibliography

[Astfalk 92] Astfalk, G , I Lustig, R Marsten and D Shanno (1992)
"The Interior Point Method for Linear Programming",
IEEE Software Review, July 1992, pp 81-86

[Astrom 95] Astrom, K J and T Hagglund (1995)
"Automatic Tuning of PID Controllers", (2nd Ed)
Instrument Society of America

[Balas 94] Balas, G J , J C Doyle, K Glover, A Packard and R Smith (1994)
"The μ-Analysis and Synthesis Toolbox for PC-MATLAB"
(PC-MATLAB is a trademark of The MathWorks Inc , MA, USA)

[Barmish 90] Barmish, B R , P P Khargonekar, Z Shi and R Tempo (1990)
"Robustness Margin need not be a Continuous Function of the Problem Data",
Systems and Control Letters, Vol 15, pp 91-98

[Beck 91] Beck, C (1991)
"Computational Issues in Solving LMIs",
Proc 30th Conf on Decision and Control, pp 1259-1260, London

[Beran 95] Beran, E , (1995)
"Induced Norm Control Toolbox",
Proc EURACO Workshop - Recent Results in Robust and Adaptive Control, Florence, Italy

[Bland 77] Bland, R. G , (1977)
"New Finite Pivoting Rules for the Simplex Method",
Mathematics of Operations Research, Vol 2, pp 103-107

[Boyd 92] Boyd, S P and C H Barratt, (1992)
"Linear Controller Design"
Prentice Hall

[Boyd 94] Boyd, S and E Vandenberghe (1994)
"Semidefinite Programming Toolbox for MATLAB"
(MATLAB is a trademark of The MathWorks Inc , MA, USA)
This code is publicly available on Stanford's FTP Server

[Bosgra 94] Bosgra, O H , P F Lambrecht and M Steinbuch (1994)
"Review of the μ-Analysis and Synthesis Toolbox (μTools)",
Automatica, Vol 30, No 4, pp 733-735

[Checkoway 92] Checkoway, C , K Kirk, D Sullivan and M Townsend, (1992)
"Simulink User's Guide for MATLAB"
(MATLAB is a trademark of The MathWorks Inc , MA, USA)

[Dahleh 88] Dahleh, M A and J B Pearson, (1988)
"Optimal Rejection of Persistent Disturbances, Robust Stability, and Mixed Sensitivity Minimization",
IEEE Transactions on Automatic Control, Vol AC-33, pp 722-731

[Dahleh 93] Diaz-Bobillo, I J and M A Dahleh, (1993)
"Minimization of the Maximum Peak-to-Peak Gain The General Multiblock Problem",
IEEE Transactions on Automatic Control, Vol AC-38, pp 1459-1482

[Dahleh 95] Dahleh, M A and I J Diaz-Bobillo (1995)
"Control of Uncertain Systems, a Linear Programming Approach"
Prentice Hall

[Dantzig 63] Dantzig, G (1963)
"Linear Programming and Extensions"
Princeton University Press

[De Carvalho 93] De Carvalho, J L M (1993)
"Dynamical Systems and Automatic Control"
Prentice Hall

[De Gaston 88] De Gaston, R R E and M G Safonov (1988)
"Exact Calculation of the Multiloop Stability Margin",
IEEE Transactions on Automatic Control, Vol 33, No 2, pp 156-171

[Demmel 92] Demmel, J (1992)
"The Componentwise Distance to the Nearest Singular Matrix",
SIAM J Matrix Analysis and Applications, Vol 13, No 1, pp 10-19

[Dines 43] Dines, L L (1943)
'On Linear Combinations of Quadratic Forms",
Bull Amer Math Soc, Vol 49, pp 388-393

[Doyle 82] Doyle, J C (1982)
"Analysis of Feedback Systems with Structured Uncertainties",
Proc IEE, Part D, Vol 129, No 6, pp 242-250

[Doyle 83] Doyle, J C (1983)
"Synthesis of Robust Controllers and Filters",
Proc IEEE Conf on Decision and Control, pp 109-114

[Doyle 87] Doyle, J C (1987)
"A Review of μ for Case Studies in Robust Control",
Proc IFAC 10th World Congress, pp 365-372, Munich

[Doyle 91] Doyle, J C , A Packard and K Zhou (1991)
"Review of LFTs, LMIs and μ ",
Proc 30th Conf on Decision and Control, London

[Doyle 92] Doyle, J C , B A Francis and A Tannenbaum (1992)
"Feedback Control Theory"
Maxwell-Macmillan Press

[El Ghaoui 91] El Ghaoui, L and P A Bliman (1991)
"Factorisation and Smallest Norm Roots of Multivariable Polynomials in Robustness Margin Calculation with Uncertain Correlated Parameters",
Proc IEEE Conf on Decision and Control, pp 19-24 Brighton, UK

[Fan 86] Fan, M K H and A L Tits (1986)
"Characterization and Efficient Computation of the Structured Singular Value",
IEEE Transactions on Automatic Control, Vol 31, pp 734-743

[Fan 2 86] Fan, M K H (1986)
"Characterization and Efficient Computation of the Structured Singular Value"
Ph D Dissertation, Maryland Institute of Technology

[Fan 88] Fan, M K H and A L Tits (1988)
"m-Form Numerical Range and the Computation of the Structured Stability Margin",
IEEE Transactions on Automatic Control, Vol 33, pp 284-289

[Fan 91] Fan, M K H , A L Tits and J C Doyle (1991)
"Robustness in the Presence of Joint Parametric Uncertainty and Unmodeled Dynamics",
IEEE Transactions on Automatic Control, Vol 36, pp 25-38

[Fan 95] Tits, A L and M K H Fan (1995)
"On the Small-μ Theorem",
Automatica, Vol 31, pp 1199-1201

[Feron 92] Feron, E , L El Ghaoui, V Balakrishnan and S Boyd (1992)
"Computing Bounds for the Structured Singular Value via an Interior Point Algorithm",
Proc American Control Conference, pp 2195-2196

[Flannery 91] Flannery, B (1991)
"Numerical Recipes in C"
Cambridge University Press

[Ford 90] Ford, M P , J M Maciejowski and J M Boyle (1990)
"Multivariable Frequency Domain Toolbox for PC-MATLAB"
(PC-MATLAB is a trademark of The MathWorks Inc., MA, USA)

[Franklin 86] Franklin, G F , J D Powell and A Emami-Naeini (1986)
"Feedback Control of Dynamic Systems"
Addison-Wesley Publishing Company

[Gantmacher 60] Gantmacher, F R (1960)
"Matrix Theory", Vol 2
Chelsea Publishing Company, NY

[Golub 89] Golub, G H and C F Van Loan (1989)
"Matrix Computations", (2nd Ed)
John Hopkins Press

[Green 94] Green, M and D J N Limebeer (1994)
"Linear Robust Control"
Prentice Hall

[Hayes 94] Hayes, M J and A M Holohan (1994)
"A New Algorithm for Robustness Analysis via the Structured Singular Value μ",
IFAC Symposium on Robust Control, Bratislava

[Hayes 94 2] Hayes, M J , P J Naughter and A M Holohan (1994)
"Worst Case Filter Analysis with Uncertain Component Values",
Irish Digital Signal Processing and Control Conference, Dublin

[Hayes 96] Hayes, M J and A M Holohan (1996)
"High Performance Control of Poorly Modelled Systems Using PID Feedback",
Irish Digital Signal Processing and Control Conference, Dublin

[Holohan 94] Holohan, A M (1994)
"A Tutorial on μ-Analysis",
Proc Euraco Network Workshop, Trinity College Dublin

[Holohan 97] Holohan, A M (1997)
"A Tutorial on μ-Analysis Part 2",
To Appear

[Kaplan 95] Kaplan, D T and L Glass (1995)
"Understanding Nonlinear Dynamics"
Springer-Verlag Press

[Karmarkar 84] Karmarkar, N (1984)
"A new Polynomial Time Algorithm for Linear Programming",
Combinatorica, Vol 4, No 8, pp 373-395

[Lay 82] Lay, S R (1982)
"Convex Sets and their Applications"
John Wiley Press

[Luenberger 69] Luenberger, D G (1969)
"Optimization by Vector Space Methods"
John Wiley Press

[Luenberger 73] Luenberger, D G (1973)
"Linear and Nonlinear Programming"
Addison-Wesley

[MacFarlane 79] MacFarlane, A G J (1979)
"Frequency Response Methods in Control Systems"
IEEE Press

[Maciejowski 89] Maciejowski, J M (1989)
"Multivariable Feedback Design"
Addison Wesley

[Morari 89] Morari, M and E Zafiriou (1989)
"Robust Process Control"
Prentice Hall

[Newlin 91] Newlin, M P and R S Smith (1991)
"Model Validation and a Generalisation of μ ",
Proc 30th Conf on Decision and Control, pp 1257-1258, London

[Osborne 60] Osborne, E E (1960).
"On Preconditioning of Matrices",
J Assoc Comp Mach, Vol 7, pp 338-345

[Packard 91] Packard, A and P Pandey (1991)
"Continuity Properties of the Real/Complex Structured Singular Value",
IEEE Transactions on Automatic Control, Vol 38, No 3, pp 415-428

[Packard 93] Packard, A and J C Doyle (1993)
"The Complex Structured Singular Value",
Automatica, Vol 29, No 1, pp 71-109

[Packard 93 2] Packard, A , J C Doyle and G Balas (1993)
"Linear, Multivariable Robust Control with a μ Perspective",
Journal of Dynamic Systems, Measurement, and Control, Vol 115, pp 426-438

[Perez 95] Perez, O , F Tadeo and P Vega (1995)
"Robust Control of a pH Control Plant",
Proc of CCA, Albany, NY

[Qiu 95] Qiu, Li , B Bernhardsson, A Rantzer, E J Davison, P M Young and J C Doyle (1995)
"A Formula for Computation of the Real Stability Radius",
Automatica, Vol 31, No 6, pp 879-890

[Radford 96] Radford, T (1996)
"And it wasn't even insured",
The Guardian, Front Page, June 5, 1996

[Safonov 82] Safonov, M G (1982)
"Stability Margins of Diagonally Perturbed Multivariable Systems",
Proc IEE, Part D, Vol 129, No 6, pp 251-256

[Safonov 81] Safonov, M G and M Athans (1981)
"A Multiloop Generalization of the Circle Criterion for Stability Margin Analysis",
IEEE Transactions on Automatic Control, Vol 26, pp 415-422

[Safonov 88] Safonov, M G and R Y Chiang (1988)
"Robust Control Toolbox for use with MATLAB"
(MATLAB is a trademark of The MathWorks Inc , MA, USA)

[Skogestad 88] Skogestad, S M Morari and J C Doyle (1988)
"Robust Control of Ill Conditioned Plants High Purity Distillation",
IEEE Transactions on Automatic Control, Vol 33, pp 1092-1105

[Smith 90] Smith, D , M Eggen and R St Andre (1990)
"A Tansition to Advanced Mathematics", (3rd Ed)
Brooks Cole Press

[Spivey 70] Spivey, W A and R M Thrall (1970)
"Linear Optimisation"
Holt, Rinehart and Winston Press

[Stein 81] Doyle, J C , and G Stein (1981)
"Multivariable Feedback Design Concepts for a Classical/Modern Synthesis",
IEEE Transactions on Automatic Control, Vol AC-26, pp 4-16

[Stem 82] Doyle, J C , J E Wall and G Stein (1982)
"Performance and Robustness Analysis for Structured Uncertainty",
Proc Conf on Decision and Control, pp 629-636, Orlando FL

[Tadeo 96] Tadeo, F (1996)
Private Communication

[Tadeo 97] Tadeo, F , A M Holohan and P Vega (1997)
"Design of a 2 DOF l_1-Optimal Controller for a pH Control Plant",
To Appear

[Tekawy 89] Tekawy, J A , M G Safonov and R Y Chiang (1989)
"Algorithms for Computing the Multivariable Stability Margin",
Proc Conf on Aerospace and Computational Control, Oxnard, USA

[Young 90] Young, P M and J C Doyle (1990)
"Computation of μ with Real and Complex Uncertainties",
Proc 29th Conf on Decision and Control, pp 1230-1235

[Young 95] Young, P M , M P Newlin and J C Doyle (1995)
"Computing Bounds for the Mixed μ Problem",
Int J of Robust and Nonlinear Control, Vol 5, pp 573-590

[Ziegler 42] Ziegler, J G and N B Nichols (1942)
"Optimum Settings for Automatic Controllers",
Trans ASME, pp 759-768